



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

**“ANÁLISIS DE LA PLATAFORMA RIA GWT PARA DESARROLLO EN AJAX
PARA EL DEPARTAMENTO DE RECURSOS HUMANOS DE LA REFINERÍA
ESTATAL DE ESMERALDAS”**

TESIS DE GRADO

**PREVIA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS
INFORMÁTICOS**

HENRY FABRICIO MURILLO VÉLEZ

MIRIAN SUSANA RIERA IDROVO

RIOBAMBA – ECUADOR

- 2010 -

AGRADECIMIENTO

Agradezco primeramente a Dios por haberle dado a mi madre Luz María Vélez Zambrano toda la fuerza y fortaleza para poderme ayudar en todo lo que he necesitado, luego a mi madre la mujer que siempre se preocupó por mí y dio hasta lo que no tuvo por sacarme adelante en esta meta que me tracé, a mis hermanos, a la Escuela Superior Politécnica de Chimborazo, a la Escuela de Ingeniería en Sistemas, docentes y amigos. De manera especial al Ing. Wladimir Castro director de nuestra tesis y al Ing. Danilo Pastor miembro del tribunal que nos preside, quienes han encaminado la presente investigación. Además agradezco la colaboración al Ing. Jorge Huilca, a Carlos Rodríguez, a Miguel Baldeon y a todos mis amigos que de una u otra forma me apoyaron en lo que más necesité.

Henry Murillo Vélez

A Dios por darme la vida, a mis padres Rosa y Julio, hermanos(as) Julio, Narcisa, Giovanny, Mercy, Carlos por el apoyo incondicional que me brindan día a día, a la Escuela Politécnica Superior de Chimborazo, a la Escuela de Ingeniería en Sistemas donde los maestros impartieron sus conocimientos, al Ing. Wladimir Castro, Ing. Danilo Pástor e Ing. Jorge Huilca quienes nos encaminaron en el momento del desarrollo de la tesis, y el apoyo de Carlos Rodríguez, Miguel Baldeón, Angel Apina, Ing. Mirian Cisneros personal de Petroecuador por ayudarnos con la información de Refinería Estatal de Esmeraldas, a mis amistades que me han tendido la mano en los momentos buenos y malos, y a mi compañero de tesis.

Mirian Riera Idrovo

DEDICATORIA

La presente tesis está dedicada principalmente a Nuestro Creador quien nos proporcionó la vida e inteligencia, así también a mi querida madrecita Luz María Vélez Zambrano quien siempre estuvo ahí para brindarme su apoyo y sacrificio ya que sin eso no hubiera tenido la dicha de conseguir esta meta tan anhelada, a mi tía Yolanda que ha sido como otra madre más para mí al igual que Doña Aida, a mis hermanos René, Joffre, Jhonny y Cristhian, a mi cuñada Johana que siempre me apoyó de una u otra forma, a mi enamorada Andrea que igual me ayudó en muchas cosas mientras realizaba la tesis, a mi amiga Maggi y su enamorado Miguel, a mi compañera de tesis y a mis amigos y para todas aquellas personas que creyeron y también para las que no creyeron en mí.

Henry Murillo Vélez

A mis padres y hermanos(as) por el amor y apoyo incondicional, a mis amistades que son parte de mi vida, que me tuvieron paciencia durante el desarrollo de mi tesis y en especial al ser misericordioso Dios.

Mirian Riera Idrovo

FIRMAS RESPONSABLES Y NOTAS

ING. IVAN MENES

**DECANO DE LA FACULTAD DE
INFORMÁTICA Y ELECTRÓNICA**

ING. RAUL ROSERO

**DIRECTOR DE LA ESCUELA
DE INGENIERÍA EN SISTEMAS**

ING. WLADIMIR CASTRO

DIRECTOR TESIS

ING. DANILO PASTOR

MIEMBRO DE TESIS

LCDO. CARLOS RODRIGUEZ

**DIRECTOR DEL CENTRO
DE DOCUMENTACIÓN**

RESPONSABILIDAD DEL AUTOR

Nosotros, Henry Fabricio Murillo Vélez y Mirian Susana Riera Idrovo, somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.

Henry Fabricio Murillo Vélez

Mirian Susana Riera Idrovo

ABREVIATURAS

AJAX:	Asynchronous JavaScript And XML - JavaScript y XML Asíncrono
API:	Application Programming Interface – Interfaz de Programación de Aplicaciones
AS/400:	Microcomputadoras
AWT:	Abstract Window Toolkit - Juego de Herramientas de Ventana Abstracta
BPEL:	Tecnologías SOA
CSS:	Cascading Style Sheets - Hojas de Estilos en Cascada
DB:	Database - Base de Datos
DB2/400:	Integrated Data Base - Base de Datos Integrada
DBMS:	Database Management System - Sistema de Gerencia de la Base de Datos
DDS:	Data Description Specification - Especificaciones de la Descripción de los Datos
DHTML:	Dynamic HTML - HTML Dinámico
DOM:	Document Object Model - Modelo de Objetos del Documento
EBML:	Extensible Binary Meta Language - Meta Lenguaje Binario Extendible
ECJ:	Compilador que viene con Eclipse
FODA:	Fortalezas, Oportunidades, Debilidades, Amenazas
GCj:	Integración del compilador de Java
GEF:	Graphic Editing Framework - Framework para la Edición Gráfica
GNU GPL:	General Public License - Licencia Pública General de GNU
GUI:	Graphic User Interface - Interfaz Gráficas de Usuario

GWT:	Google Web Toolkit
HTML:	HyperText Markup Language – Lenguaje de Etiquetado de Hipertexto
HTTP:	Hypertext Markup Language - Protocolo de Transferencia de Hipertexto
IBM:	International Bussiness Machines – Máquinas de Negocio Internacional
IDE:	Integrated Developement Environment - Entorno de Desarrollo Integrado.
IFS:	Integrated File System - Sistema de Ficheros Integrado
IMS:	Information Management System - Sistema de Gerencia de Información
J2EE:	Java 2 Enterprise Edition
JDBC:	Java DataBase Conectivity - Conectividad de Bases de Datos con Java
JDK:	Java Development Kit - Kit de Desarrollo de Java.
JDT:	Java Development Toolkit - Herramientas de Desarrollo Java
JNI:	Java Native Interface - Interfaces Nativas de Java.
JRE:	Java Runtime Enviroment - Entorno en tiempo de Ejecución de Java.
JS:	Java Script
JSF:	Java Server Faces
JSNI:	JavaScript Native Interface - Interfaces Nativas de Java Script.
JSO:	Java Script Object – Objeto Java Script
JSON:	JavaScript Object Notation – Notación de Objetos de JavaScript
JSP:	JavaServer Pages
JVM:	Java Virtual Machine - Máquina Virtual de Java
LAMP:	Linux + Apache + Mysql + PHP
LIC:	Licensed Internal Code - Código Interno Licenciado
MI:	Machine Interface - Interfaz de la Máquina

MSRC:	Microsoft's Remote Scripting - Scripting Remoto de Microsoft
NBM:	NetBeans Module – Módulo Netbeans
OO:	Orientado a Objetos.
RAD:	Rational Application Developer
REE:	Refinería Estatal de Esmeraldas
RIA:	Rich Internet Applications - Aplicaciones Ricas en Internet
RMI:	Remote Mechanism Method Invocation - Mecanismo de Invocación de Métodos Remotos.
RPC:	Remote Procedure Service - Llamadas a Procedimientos Remotos
SDK:	Software Development Kit - Kit de Desarrollo de Software
SISREHU:	Sistema de Recursos Humanos
SOA:	Service Oriented Architecture – Arquitectura Orientada a Servicios
SOAP:	Simple Object Access Protocol – Protocolo de Acceso a Objetos Simples
SPA:	Services Paradigm Adoption - Servicios de Adopción Paradigma
SQL:	Structured Query Language - Lenguaje de Interrogación Estructurado
SRS:	Especificación de Requerimientos de Software.
SWT:	Standard Widget Toolkit
UML:	Unified Modeling Language - Unificado de Modelado Idioma
URL:	Uniform Resource Locator – Localizador Uniforme de Recursos
WSAD:	WebSphere Studio Application Developer
XHTML:	eXtensible HyperText Markup Language - Lenguaje de Etiquetado de Hipertexto Extensible
XML:	Extended Markup Language – Lenguaje de Marcado Extendido

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

FIRMAS RESPONSABLES Y NOTAS

RESPONSABILIDAD DEL AUTOR

ABREVIATURAS

INDICE DE TABLAS

INDICE DE GRÁFICOS

INTRODUCCIÓN

CAPÍTULO I MARCO REFERENCIAL	27
1.1. Antecedentes	27
1.2. Justificación	28
1.2.1. Justificación teórica	28
1.2.2. Justificación práctica	30
1.3. Objetivos.....	32
1.3.1. Objetivo general	32
1.3.2. Objetivos específicos.....	32
1.4. Hipótesis.....	33
1.5. Métodos y técnicas	33
1.5.1. Métodos.....	33
1.5.2. Técnicas.....	35
CAPÍTULO II MARCO TEORICO	36
2.1. Introducción.....	36
2.2. Aspectos generales de Java.....	37
2.2.1. Lenguaje Java.....	37
2.2.1.1. Definición	37

2.2.1.2. Características del lenguaje.....	38
2.2.1.3. Máquina Virtual Java.....	39
2.2.1.4. JRE.....	40
2.2.1.5. JDK.....	40
2.2.2. Aplicación web en Java	41
2.2.2.1. Aplicaciones web	41
2.2.2.2. Aplicaciones web en Java	41
2.3. Aspectos generales de AJAX.....	43
2.3.1. Introducción	43
2.3.2. Definición.....	44
2.3.3. Antecedentes.....	45
2.3.3.1. OpenAjax Alliance	46
2.3.3.2. Modelo AJAX de Aplicaciones Web	47
2.3.4. Importancia de AJAX.....	48
2.3.5. Beneficios.....	49
2.3.6. Ventajas.....	49
2.3.7. Desventajas	50
2.4. ENTORNOS DE DESARROLLO INTEGRADOS (IDE's).....	51
2.4.1. Tipos de IDEs	51
2.4.1.1. Descripción de IDEs.....	51
2.5. Plataforma RIA-GWT	54
2.5.1. Plataforma RIA (Rich Internet Application).....	54
2.5.1.1. Historia.....	54
2.5.1.2. Ventajas de la Plataforma RIA.....	55
2.5.1.3. Desventajas de la Plataforma RIA.....	56
2.5.2. Framework GWT	57

2.5.2.1. Introducción	57
2.5.2.2. Definiciones de GWT	59
2.5.2.3. Características de GWT	60
2.5.2.4. Arquitectura GWT	63
2.5.2.5. Para qué sirve el framework GWT	66
2.5.2.6. Funciones y Funcionamiento de GWT	67
2.5.2.6.1. Funciones	67
2.5.2.6.2. Funcionamiento	69
2.5.2.7. Ventajas de GWT	70
2.5.2.8. Desventajas de GWT	70
2.5.2.9. Ciclo de desarrollo.....	71
2.5.2.10. Modelo de programación en GWT.....	72
2.5.2.11. Estructura de un proyecto GWT.....	74
2.5.2.12. Los módulos en GWT.....	78
2.5.2.12.1. Formatos de los módulos XML.....	80
2.5.2.12.2. Inclusión automática de recursos.....	83
2.5.2.12.3. Filtrando paquetes públicos.....	84
2.5.2.13. Herramientas de la línea de comandos.....	85
2.5.2.13.1. projectCreator	85
2.5.2.13.2. applicationCreator	85
2.5.2.13.3. junitCreator	86
2.5.2.13.4. i18nCreator.....	86
2.5.2.14. Construcción de Interfaces de Usuarios GUI.....	86
2.5.2.14.1. Widget y paneles	87
2.5.2.14.2. Events y Listeners.....	115
2.5.2.14.3. Creando widget personalizados	117

2.5.2.14.4. Hojas de estilos (CSS)	119
2.5.2.14.5. Atado de imágenes (image bundle)	121
2.5.2.15. RPC (Remote Procedure Call)	123
2.5.2.15.1. Definición de RPC.....	124
2.5.2.15.2. Ventajas del RPC-GWT.....	125
2.5.2.15.3. Desventaja del RPC-GWT	125
2.5.2.15.4. Anatomía de RPC en GWT.....	126
2.5.2.15.5. Crear servicios RPC.....	132
2.5.2.15.6. Invocar servicios.....	135
2.5.2.16. Desplegar en Tomcat	137
CAPÍTULO III ANÁLISIS COMPARATIVO	139
3.1. Introducción	139
3.2. Determinación de las herramientas a comparar.....	140
3.3. Análisis de las herramientas.....	146
3.3.1. Eclipse.....	146
3.3.1.1. Introducción	146
3.3.1.2. Definición	147
3.3.1.3. Arquitectura del IDE Eclipse	147
3.3.1.4. Estructura del IDE Eclipse.....	151
3.3.1.5. Características	153
3.3.1.6. El proyecto Eclipse.....	154
3.3.1.7. Sub-proyectos de Eclipse	154
3.3.1.8. La librería SWT.....	155
3.3.2. Netbeans.....	156
3.3.2.1. Introducción	156
3.3.2.2. Definición	157

3.3.2.3. Características	158
3.3.2.4. Tipos de proyectos.....	162
3.3.2.5. Proceso de desarrollo.....	166
3.3.2.6. Ventajas.....	166
3.3.2.7. Módulos Integrados de Netbeans	167
3.3.2.8. Paquetes de Netbeans IDE	169
3.3.3. INTELLIJ IDEA.....	172
3.3.3.1. Introducción	172
3.3.3.2. Definición	172
3.3.3.3. Características	172
3.3.3.4. Ventajas.....	173
3.3.3.5. Función.....	174
3.4. Determinación de los parámetros de comparación.....	174
3.4.1. Parámetro 1: Acceso a la Base de Datos.....	175
3.4.2. Parámetro 2: Compilación	175
3.4.3. Parámetro 3: Interfaz Gráfica de Usuario (GUI).....	175
3.4.4. Parámetro 4: Seguridad.....	175
3.4.5. Parámetro 5: Soporte Técnico de GWT	176
3.5. Determinación de las variables para los parámetros de comparación	176
3.5.1. Parámetro 1: Acceso a la Base de Datos.....	176
3.5.2. Parámetro 2: Compilación	176
3.5.3. Parámetro 3: Interfaz Gráfica de Usuario (GUI).....	177
3.5.4. Parámetro 4: Seguridad.....	177
3.5.5. Parámetro 5: Soporte Técnico de GWT	177
3.6. Análisis comparativo	177
3.6.1. Parámetro 1: Acceso a la Base de Datos.....	180

3.6.1.1. Determinación de Variables.....	180
3.6.1.2. Valoraciones.....	180
3.6.1.3. Interpretación	181
3.6.1.4. Calificación	182
3.6.2. Parámetro 2: Compilación	183
3.6.2.1. Determinación de Variables.....	183
3.6.2.2. Valoraciones.....	183
3.6.2.3. Interpretación	184
3.6.2.4. Calificación	185
3.6.3. Parámetro 3: Interfaz Gráfica de Usuario (GUI).....	185
3.6.3.1. Determinación de Variables.....	185
3.6.3.2. Valoraciones.....	186
3.6.3.3. Interpretación	187
3.6.3.4. Calificación	188
3.6.4. Parámetro 4: Seguridad.....	188
3.6.4.1. Determinación de Variables.....	188
3.6.4.2. Valoraciones.....	189
3.6.4.3. Interpretación	190
3.6.4.4. Calificación	190
3.6.5. Parámetro 5: Soporte Técnico de GWT.....	191
3.6.5.1. Determinación de Variables.....	191
3.6.5.2. Valoraciones.....	191
3.6.5.3. Interpretación	192
3.6.5.4. Calificación	193
3.7. Puntajes alcanzados.....	193
3.8. Interpretación	196

3.9.	Resultado del análisis	196
3.10.	Conclusión	199
3.11.	Demostración de la hipótesis.....	199
3.11.1.	Tiempos de Respuesta SIN la plataforma RIA GWT	200
3.11.1.1.	Módulo de Consultas Generales	200
3.11.2.	Módulo de Estadísticas	203
3.11.3.	Tiempos de Respuestas CON la plataforma RIA GWT	205
3.11.3.1.	Módulo de Consultas Generales	205
3.11.3.2.	Modulo de Estadísticas visualizado en barras.	207
3.11.4.	Análisis matemático	209
3.11.4.1.	Módulo de Consultas Generales	210
3.11.4.2.	Módulo de Estadísticas.....	215
3.11.5.	Interpretación de los Resultados.....	219
3.11.6.	Conclusión	222
CAPÍTULO IV METODOLOGÍA DE ANÁLISIS		223
4.1.	Introducción	223
4.2.	Concepto FODA.....	224
4.3.	Componentes del FODA.....	225
4.4.	Diagnóstico de luz y Fuerza del Centro (LFC)	226
4.5.	Propuesta metodológica	226
4.6.	Desarrollo de las fases de la metodología	227
4.6.1.	Breve descripción de las fases.....	228
4.6.1.1.	Análisis Estratégico	228
4.6.1.2.	Ampliación del análisis estratégico	228
4.6.1.3.	Ponderación de las estrategias.....	228
4.6.1.4.	Interpretación de los resultados.....	229

4.6.2. Análisis Estratégico	229
4.6.2.1. Análisis Interno	229
4.6.2.1.1. Fortalezas.....	229
4.6.2.1.2. Debilidad	232
4.6.2.2. Análisis Externo	234
4.6.2.2.1. Amenazas	234
4.6.2.2.2. Oportunidades.....	235
4.6.3. Ampliación del análisis estratégico	236
4.6.3.1. Criterios de valoración de la matriz de confrontación	236
4.6.3.2. Elaboración de la matriz de confrontación.....	236
4.6.3.3. Elaboración de la matriz FODA modificada.....	237
4.6.4. Diseño de las opciones estratégicas.....	238
4.6.4.1. Elección de la clasificación de estrategias	238
4.6.4.2. Despliegue de las estrategias.....	240
4.6.4.3. Elaboración de la matriz de estrategias.....	241
4.6.5. Ponderación de las estrategias.....	243
4.6.5.2. Jerarquización de Estrategias Defensivas (ED).....	244
4.6.6. Interpretación de resultados	245
CAPÍTULO V DESARROLLO DE LA APLICACIÓN WEB PARA LOS EMPLEADOS DE	
REFINERIA ESTATAL DE ESMERALDAS.....	247
5.1. Introducción	247
5.2. Instalación de GWT.....	248
5.2.1. Prerrequisitos para instalar GWT	248
5.2.2. Plugin utilizados	248
5.3. Configuración de GWT en Eclipse.....	251
5.4. Conexión con Base de Datos.....	252

5.4.1. Conexiones con AS/400.....	252
5.4.2. JDBC	252
5.4.3. Cargar el controlador JDBC.....	253
5.5. Desarrollo de un proyecto GWT	256
5.5.1. Crear un proyecto GWT.....	256
5.5.2. Configuración de la Librería GWT.....	258
5.5.3. Creación de un módulo en GWT.....	259
5.5.4. Ejecución del Código.....	263
5.6. Documentación Técnica de la Aplicación Web “SISREHU”	265
5.6.1. Planificación y Análisis: Documento SRS.....	265
5.6.1.1. Introducción	265
5.6.1.2. Riesgos.....	266
5.6.1.2.1. Listado de Riesgos	266
5.6.1.2.3. Análisis de Riesgos	268
5.6.1.2.4. Resultados.....	268
5.6.1.3. Requerimientos Funcionales	269
5.6.1.3.1. Actores	269
5.6.1.3.2. Diagramas de Casos de Uso	269
5.6.1.4. Casos de Uso.....	273
5.6.1.4.1. Caso de Uso: Información General.....	273
5.6.1.4.2. Caso de Uso: Iniciar Sesión.....	273
5.6.1.4.3. Caso de Uso: Cerrar Sesión	274
5.6.1.4.4. Caso de Uso: Datos Generales (Asistencia, Cursos Recibidos, Datos Familiares, Datos Personales, Historial Financiero, Record Académico y Saldo Vacacional).....	275
5.6.1.4.5. Caso de Uso: Desempeño Laboral	276
5.6.1.4.6. Caso de Uso: Estadísticas	276

5.6.1.5. Requerimientos Detallados	277
5.6.1.5.1. Modulo 1: INFORMATIVO.....	278
5.6.1.5.2. Modulo 2: NOTICIAS	278
5.6.1.5.3. Modulo 3: AUTENTICACIÓN.....	278
5.6.1.5.4. Modulo 4: CONSULTAS GENERALES.....	278
5.6.1.5.5. Modulo 5: ESTADISTICAS.....	279
5.6.1.6. Requerimientos del Sistema.....	279
5.6.1.7. Diccionario de Datos	280
5.6.2. Diseño.....	284
5.6.2.1. Diseño de la Base de Datos de la Aplicación Web “SISREHU”	284
5.6.2.2. Diagrama de Clases de la Aplicación Web “SISREHU”.....	285
5.6.2.3. Diagrama de Despliegue de la Aplicación Web “SISREHU”.....	286
5.6.2.4. Diagrama de Componentes de la Aplicación Web “SISREHU”.....	287
5.6.3. Pruebas y Validaciones.....	287
5.6.3.1. Pruebas.....	288
5.6.3.2. Validaciones.....	288
CONCLUSIONES	
RECOMENDACIONES	
RESÚMEN	
SUMMARY	
ANEXOS	
GLOSARIO	
BIBLIOGRAFÍA	

INDICE DE TABLAS

Tabla II.1. Estructura de proyecto.....	74
Tabla II.2: Tipos de datos que usa RPC	128
Tabla II.3: Ejemplo del servicio Async.....	132
Tabla III.1: Resultados IDE Adecuado	145
Tabla III.2: Escala de puntuación para calificación de parámetros	178
Tabla III.3: Escala de valoración cuantitativa para los parámetros.....	178
Tabla III.4: Variable Acceso a la Base de Datos	181
Tabla III.5: Variable Compilación.....	184
Tabla III.6: Variable Interfaz Gráfica de Usuario (GUI)	187
Tabla III.7: Variable Seguridad	189
Tabla III.8 Variable Soporte Técnico con respecto a GWT	192
Tabla III.9: Tabla General de Resultados.....	194
Tabla IV.1: Criterios de valoración de la matriz de confrontación.....	236
Tabla IV.2: Matriz de confrontación.....	236
Tabla IV.3: Matriz FODA modificada.....	237
Tabla IV.4: Elección de la clasificación de estrategias	239
Tabla IV.5: Despliegue de las estrategias	240
Tabla IV.6: Matriz de estrategias.....	242
Tabla IV.7: Jerarquía de Estrategias de Supervivencia (ES).....	243
Tabla IV.8: Jerarquía de Estrategias Defensivas (ED).....	244
Tabla V.1: Lista de Riesgos.....	266
Tabla V.2: Determinación de la Probabilidad	266
Tabla V.3: Determinación del Impacto	267
Tabla V.4: Determinación de la Exposición del Riesgo	267
Tabla V.5: Equivalencia de riesgos	267

Tabla V.6: Análisis de Riesgo	268
Tabla V.7: Resultado del Análisis de Riesgo	268
Tabla V.7: Actores de Caso de Uso	269
Tabla V.9: Caso de Uso de Información General	273
Tabla V.10: Caso de Uso de Iniciar Sesión	273
Tabla V.11: Caso de Uso de Cerrar Sesión	274
Tabla V.12: Caso de Uso de Datos Generales	275
Tabla V.13: Caso de Uso de Desempeño Laboral	276
Tabla V.14: Caso de Uso de Estadísticas	276
Tabla V.15: Requerimientos del Sistema	279
Tabla V.16: Diccionario de Datos	280

INDICE DE GRAFICOS

Gráfico II.1: Modelos comparativos de aplicaciones Web.....	47
Gráfico II.2: Modelos clásico de aplicaciones Web.....	47
Gráfico II.3: Modelos AJAX de aplicaciones Web.	48
Gráfico II.4: Visión General de la historia de las aplicaciones de software.....	55
Gráfico II.5: Arquitectura de GWT.....	63
Gráfico II.6: Compilador GWT Java-a-JavaScript	64
Gráfico II.7: Navegador Web “Hosted” de GWT.....	64
Gráfico II.8: Librería de clases de interfaz de usuario de GWT	66
Gráfico II.9: Funcionamiento de GWT	69
Gráfico II.10: Ejemplo de un widget (botón)	88
Gráfico II.11: Jerarquía de clases de los widgets GWT.....	91
Gráfico II.12: Categorías de los Widgets	93
Gráfico II.13: Apariencia del Widget Label.....	94
Gráfico II.14: Apariencia de un label al aplicarle css	94
Gráfico II.15: Apariencia del Widget HTML.....	95
Gráfico II.16: Apariencia del Widget Image	96
Gráfico II.17: Apariencia del Widget Image con URL externa.....	97
Gráfico II.18: Ejemplo del código de uso del Hyperlink	98
Gráfico II.19: Ejemplo gráfico del uso del Hyperlink	99
Gráfico II.20: Creación del Widget Button	101
Gráfico II.21: Modo gráfico del widget Button.....	101
Gráfico II.22: Creación del Widget ToggleButton	102
Gráfico II.23: Estado de los botones.....	103
Gráfico II.24: Creación de un widget CheckBox.....	104
Gráfico II.25: Creación de un RadioButton.....	105

Gráfico II.26: Modo gráfico del widget RadioButton.....	105
Gráfico II.27: Modo gráfico del widget ListBox	106
Gráfico II.28: Creación de un ListBox (primera manera)	106
Gráfico II.29: Creación del ListBox (segunda manera)	107
Gráfico II.30: Creación del widget TextBox	107
Gráfico II.31: Ejemplo de cómo utilizar un TextBox	108
Gráfico II.32: Modo gráfico del TextBox	108
Gráfico II.33: Modo gráfico del widget PasswordTextBox	109
Gráfico II.34: Creación del widget TextArea	109
Gráfico II.35: Modo gráfico del widget TextArea.....	110
Gráfico II.36: Ejemplo del widget RichTextArea.....	110
Gráfico II.37: Modo gráfico de un FileUpload.....	111
Gráfico II.38: Creación de un TreeItem	112
Gráfico II.39: Creación del widget TreeItem	113
Gráfico II.40: Modo gráfico del widget TreeItem	113
Gráfico II.41: Creación del widget MenuBar	114
Gráfico II.42: Creación del widget MenuBar	114
Gráfico II.43: Modo gráfico del MenuBar	115
Gráfico II.44: Ejemplo de widget compuesto.....	118
Gráfico II.45: Modo gráfico del widget compuesto.....	118
Gráfico II.46: Comunicación entre el cliente y el servidor RPC	125
Gráfico II.47: Anatomía de RPC en GWT	127
Gráfico II.48: Crear un servicio RPC.....	129
Gráfico II.49: Creación del servlet para el servicioRPC	130
Gráfico II.50: Creación del Objeto Proxy	130
Gráfico II.51: Ejemplo del Objeto proxy	131

Gráfico II.52: Primera pantalla para crear un servicio RPC.....	132
Gráfico II.53: Segunda pantalla para crear un servicio RPC.....	133
Gráfico II.54: Visualización de las interfaces del servicio RPC.....	133
Gráfico II.55: Sobrescribir la clase REE_ServiceImpl	135
Gráfico II.56: Instanciar el objeto proxy	136
Gráfico II.57: Especificar la URL del servicio.....	136
Gráfico II.58: Crear un Objeto CallBack	137
Gráfico II.59: Llamar al servicio remoto.....	137
Gráfico III.1: Arquitectura de la Plataforma Eclipse	151
Gráfico III.2: Estructura del IDE Eclipse.....	153
Gráfico III.3: Módulos del IDE Netbeans.....	163
Gráfico III.4: Módulo Envoltorio de Librería del IDE Netbeans	163
Gráfico III.5: Módulo Suite del IDE Netbeans.....	164
Gráfico III.6: Deshabilitando los módulos en el IDE Netbeans	164
Gráfico III.7: Aplicación sobre el IDE Netbeans.....	165
Gráfico III.8: Habilitando los módulos del IDE Netbeans	165
Gráfico III.9: Comparación Estadística del parámetro de Acceso a la Base de Datos.....	182
Gráfico III.10: Comparación Estadística del parámetro de la Compilación.....	185
Gráfico III.11: Comparación estadística del parámetro Interfaz Gráfico de Usuario (GUI)	188
Gráfico III.12: Comparación estadística del parámetro de la seguridad	191
Gráfico III.13: Comparación estadística del parámetro Soporte Técnico con GWT	193
Gráfico III.14: Diagrama General de Resultados	195
Gráfico III.15: Módulos de las Consultas Generales sin RIA-GWT	202
Gráfico III.16: Diagrama del Módulo de Estadísticas Sin RIA GWT	205
Gráfico III.17: Módulos de las Consultas Generales con RIA-GWT	207
Gráfico III.18: Diagrama del Módulo de Estadísticas Con RIA GWT	209

Grafico IV.1: Componentes del FODA.....	225
Gráfico IV.2: Tipos de análisis del FODA.....	226
Gráfico IV.3: Etapas del proceso de análisis FODA.....	227
Gráfico V.1: Instalación GWT en la PC (Paso 4).....	250
Gráfico V.2: Instalación GWT en la PC (Paso 5).....	250
Gráfico V.3: Instalación GWT en la PC (Paso 6).....	251
Gráfico V.4: Instalación GWT en Eclipse (Paso 2).....	251
Gráfico V.5: Instalación GWT en Eclipse (Paso 3).....	252
Gráfico V.6: Pantalla para crear un proyecto web dinámico.....	257
Gráfico V.7: Llenar datos para crear un proyecto GWT.....	257
Gráfico V.8: Primera pantalla para configurar la librería GWT.....	258
Gráfico V.9: Segunda pantalla para configurar la librería GWT.....	258
Gráfico V.10: Crear el modulo GWT.....	259
Gráfico V.11: Llenar los datos del módulo GWT.....	260
Gráfico V.12: Jerarquía de paquetes.....	260
Gráfico V.13: Ejecutar una aplicación GWT con Hosted Mode.....	263
Gráfico V.14: Primera ventana del modo hosted Mode Shell Windows.....	264
Gráfico V.15: Modo hosted simulando el navegador.....	264
Gráfico V.16: Caso de Uso General.....	269
Gráfico V.17: Caso de Uso del Usuario_Invitado.....	270
Gráfico V.18: Caso de Uso del Usuario.....	271
Gráfico V.19: Caso de Uso del Usuario_Administrador.....	272
Gráfico V.20: Diseño de la Base de Datos.....	284
Gráfico V.21: Diagrama de Clases.....	285
Gráfico V.22: Diagrama de Despliegue.....	286
Gráfico V.23: Diagrama de Componentes.....	287

INTRODUCCION

Actualmente vivimos en un mundo de evolución en donde la mayoría de los procesos que se realizaban personalmente ha cambiado pues hoy en día las entidades y empresas están automatizando todo aquello utilizando el medio global más difundido que es la Internet, a través de las aplicaciones web.

En el capítulo I Marco Referencial, se plantea el estudio de una de la herramienta que está revolucionando el mundo actual en el desarrollo de aplicaciones web como es el framework GWT desarrollado por Google, el mismo que es parte de la plataforma RIA (Rich Internet Application), permitiendo escapar de la “matriz” de tecnologías usadas actualmente para escribir aplicaciones AJAX, las cuales son difíciles de manejar y propensas a errores.

En el capítulo II Marco Teórico, se realiza un estudio teórico de los conceptos necesarios para entender completamente el funcionamiento del framework GWT para lo cual se estudia toda la terminología necesaria al lenguaje de desarrollo JAVA, lo referente a AJAX, una breve descripción de los IDE's que se puede usar y todo lo referente a GWT como instalación, ventajas, desventajas, widget, etc.

En el capítulo III Análisis Comparativo, se estudia todos los IDE's que GWT puede usar para el desarrollo de las aplicaciones web usando el lenguaje de programación Java junto con AJAX, para luego realizar el análisis correspondiente en base a una serie de parámetros y variables que permitirán seleccionar el IDE más adecuado para el desarrollo de la

aplicación web para el Departamento de Recursos Humanos de la Refinería Estatal de Esmeraldas.

En el capítulo IV Propuesta Metodológica, se propone una metodología para agilizar los procesos que se realizan actualmente a los empleados de la Refinería Estatal de Esmeraldas y que permitirán reducir el tiempo para conseguir los resultados esperados en cada proceso.

En el capítulo V Ingeniería de Software, se aplica los conocimientos obtenidos en el capítulo II y se toma en consideración el IDE determinado en el capítulo III el cual permitirá realizar el desarrollo de la aplicación web para el Departamento de Recursos Humanos de la Refinería estatal de Esmeraldas.

CAPÍTULO I MARCO REFERENCIAL

1.1. Antecedentes

Rich Internet Application (RIA) es una Aplicación de Internet Enriquecida que trata con aplicaciones hechas con AJAX (es decir que no usen Flash, Silverlight o JavaFX). Entre las plataformas que utilizan esta aplicación tenemos GWT.

Google Web Toolkit (GWT) es un framework de desarrollo en Java de código abierto, que permite escapar de la “matriz” de tecnologías usadas actualmente para escribir aplicaciones AJAX, las cuales son difíciles de manejar y propensas a errores. Con GWT, se puede desarrollar y depurar aplicaciones AJAX usando el lenguaje de programación Java en el entorno de desarrollo de nuestra preferencia (es decir, el sistema operativo y los IDEs). Una vez terminada la aplicación (escrita en Java), GWT compila y traduce dicho programa a JavaScript y HTML compatible con cualquier navegador web.

Existe en la actualidad un equipo de GWT el mismo que ha creado un repositorio donde está publicado todas las aplicaciones que usan GWT, entre las aplicaciones que más utilizan todo el potencial de GWT está la de Lombardi Blueprint.

La Refinería Estatal de **Esmeraldas (REE)**, está situada en la provincia de Esmeraldas, a una distancia de 7 1/2 Km de la ciudad de Esmeraldas, vía Atacames. Fue diseñada y construida entre 1975 y 1977 para procesar 55.600 barriles de petróleo diarios. REE, como toda empresa, para cumplir con sus objetivos posee tres unidades de trabajo: administrativas, operativas y de mantenimiento.

En la actualidad la Refinería Estatal de Esmeraldas (REE), no consta con una aplicación web para el área de recursos humanos, sin embargo, los datos de ésta área residen en la plataforma AS/400 de IBM y la base de datos se encuentra en la matriz de la ciudad de Quito la misma que tiene un motor de base de datos realizada en DB2.

1.2. Justificación

1.2.1. Justificación teórica

El Google Web Toolkit (GWT), es una framework desarrollado por google de última tecnología que permite al usuario desarrollar aplicaciones basadas en java de código abierto y usando a su vez AJAX, lo cual nos va a facilitar poder cargar de una página web lo necesario como por ejemplo los elementos multimedia (videos, gráficos, etc.).

Sin embargo escribir aplicaciones en AJAX, no es un proceso simple y no existen muchas herramientas que faciliten el trabajo. Google Web Toolkit (GWT) se presenta como una solución simple y segura para desarrolladores que deseen aprovechar el potencial de AJAX. GWT ofrece clases genéricas de HTTP que se utilizan para construir una petición y clases JSON y XML que se pueden utilizar para procesar la respuesta al cliente.

Los IDEs (Integrated Development Enviroment), Entorno Integrado de Desarrollo pueden ser exclusivos para un lenguaje de programación o bien se pueden usar para varios, como ejemplo de IDEs tenemos Eclipse, NetBeans, IDEA entre otros, y para este trabajo se usará Eclipse 3.4 Ganymede por brindar mejor funcionalidad y compatibilidad para poder trabajar con GWT.

El uso de GWT es de mucha importancia porque en la actualidad según el gobierno todas las instituciones públicas deben orientarse a la utilización de sistemas operativos, programas, bases de datos, herramientas, etc. de código libre entonces por ser Refinería una de estas empresas es de vital importancia tener una aplicación web que cumplan con estas características.

A continuación tenemos varias ventajas al trabajar con GWT:

- Cualquier carga que se realice dentro de la página web se lo hará en la parte del cliente y no en el servidor permitiendo así tener mejores tiempos de respuestas para una solicitud.

- Usa el entorno de desarrollo integrado (IDE) favorito para escribir y depurar una aplicación en Java, usando las librerías GWT que se necesite.
- Emplea el compilador de Java a JavaScript, para transformar la aplicación en un conjunto de archivos JavaScript y HTML que se puede montar en cualquier servidor y ejecutarse desde un navegador web.
- Verifica que las aplicaciones trabajen sobre todos y cada uno de los navegadores que se consideran que son los más usados por los clientes.
- Las aplicaciones generadas por GWT ejecutan código Java del lado del servidor, utilizando RPC (Llamada a Procedimiento Remoto), para la comunicación entre el Cliente y el Servidor, llevando a cabo llamadas asíncronas.
- Son más los desarrolladores que conocen Java que los que conocen JavaScript o ActionScript (el lenguaje empleado por Flash), además Java tiene la suerte de tener abundancia de herramientas para el desarrollador.

1.2.2. Justificación práctica

En vista de que en la actualidad en la Refinería Estatal de Esmeraldas no se cuenta con una Aplicación Web que contenga toda la información necesaria para el control de actividades de todos los empleados que brinde a cada uno de ellos servicios que son de mucha importancia para el sector más primordial de la empresa como: consultas de saldos de vacaciones, asistencias, record académico etc.

Al desarrollar esta aplicación web se busca que los empleados de la refinería tengan comodidad en todo lo mencionado anteriormente desde cualquier lugar a través del Internet.

La aplicación web nos permitirá acceder a:

- Autenticarse al empleado para realizar alguna tarea específica.
- Información general de la Refinería Estatal de Esmeraldas (REE) como es Misión, Visión, Objetivos y datos generales.
- Noticias publicadas por la REE por cada uno de sus departamentos: Unidad de Sistemas, Contrato, Abastecimientos, etc.
- Gestión de las acciones del personal en departamento de recursos humanos.
 - Consultas de asistencia de los empleados.
 - Consultas de los cursos recibidos por los empleados en su vida laboral.
 - Consultas de los datos familiares de los empleados.
 - Consultas de los datos personales de los empleados.
 - Consultas del desempeño laboral en sus evaluaciones en una fecha determinada.
 - Consultas del historial financiero de los empleados.
 - Consultas del record académico de los empleados.
 - Consultas del saldo vacacional de los empleados.
- Modulo de Estadísticas visualizado en barras.
 - Desempeño de los empleados.

- Número de mujeres y hombres de la empresa.
- Número de empleados con nivel académico.
- Número de profesionales principales de la empresa.
- Número de trabajadores contratados en el mes.
- Número de trabajadores en las diferentes áreas de trabajo en la empresa.

Cabe recalcar que nuestra aplicación web no se vinculará con nada que tenga que ver con el departamento financiero.

1.3. Objetivos

1.3.1. Objetivo general

Analizar la plataforma RIA GWT para desarrollo en AJAX para el departamento de Recursos Humanos de la Refinería Estatal de Esmeraldas (REE).

1.3.2. Objetivos específicos

- Estudiar las características, ventajas y desventajas de la plataforma Google Web Toolkit.
- Analizar los distintos IDEs que se pueden emplear en la plataforma GWT y buscar el más adecuado para el desarrollo de nuestra aplicación.
- Elaborar una propuesta metodológica para determinar las ventajas y desventajas que se tienen al usar la plataforma de desarrollo GWT.

- Analizar los procesos que se realizan a los empleados de la Refinería Estatal de Esmeraldas (REE), para luego diseñar y desarrollar la Aplicación Web, partiendo de la plataforma GWT con su el IDEs más adecuado.
- Realizar pruebas y validaciones de la aplicación web desarrollada.

1.4. Hipótesis

Mediante el uso de la tecnología RIA GWT se reducirá el tiempo en el flujo de trabajo de los procesos que realiza el departamento de Recursos Humanos de la Refinería Estatal de Esmeraldas.

1.5. Métodos y técnicas

La presente investigación realizada y fundamentada en el paradigma cuantitativo y cualitativo de conocimientos, pretende solucionar una problemática puntual que permitirá cubrir un problema expuesto en los antecedentes. Por tal razón, los métodos y técnicas e instrumentos que nos proporciona la metodología de la investigación científica de vital importancia y transcendencia en el desarrollo del presente proyecto.

1.5.1. Métodos

Se utilizarán los métodos que nos va seguir el camino para conseguir los resultados de los objetivos planteados. Entre los métodos tenemos:

a) **Método científico.-** Este será el método que avalará nuestra investigación científica porque a su vez contempla los siguientes puntos que involucran el desarrollo de esta tesis:

- Planteamiento del problema
- Formulación de la hipótesis
- Levantamiento o recopilación de la información
- Análisis e interpretación de resultados
- Comprobación de la hipótesis
- Difusión de resultados

b) **Método analítico - sintético.-** Este método, como lo dice su nombre, es el análisis que se realizará de los aspectos delimitados de la presente investigación que permitirá conocer, comprender y estudiar el objeto de estudio en partes y del todo investigado en sus diferentes componentes para el planteamiento de la metodología propuesta por los autores. Además el mismo, es aplicado en la sistematización de la bibliografía, que será analizada para entenderla y describirla.

c) **Método analítico - crítico.-** Este método es utilizado puesto que analizamos y sintetizamos los conocimientos, basados en la sustentación de la bibliografía, plasmados en los resultados del proyecto.

- d) **Método descriptivo.-** Este método se utilizará en el momento que se describa las características de la aplicación, conjuntamente con sus interfaces gráficas, además se realizará la explicación de cada función y procedimiento que se utilice para la realización de la aplicación y por ende el cumplimiento del objetivo general que se plantea.

1.5.2. Técnicas

- a) **Encuesta.-** Serán diseñada una serie de encuestas, para obtener información atreves de un sistema de preguntas las cuales una vez aplicadas a los empleados de REE, serán utilizadas para la recopilación de la información y luego tabularla, graficarla y analizarla.
- b) **Entrevista.-** La misma que se la aplicará al Jefe de la Unidad de Sistemas de la REE, a fin de conocer las principales necesidades, características y/o dificultades que se tienen en los demás sistemas de la empresa para evitarlos en la aplicación web planteada en este trabajo.
- c) **Lluvia de ideas.-** La lluvia de ideas será una de las principales técnicas para recolectar la información y para procesarla, por el mismo hecho que esta investigación se realizará en equipo.

CAPÍTULO II MARCO TEORICO

2.1. Introducción

En el presente capítulo se realiza un estudio teórico de los conceptos necesarios que permitan entender completamente el funcionamiento del framework GWT para lo cual se estudia todo lo necesario al lenguaje de desarrollo JAVA como su definición, característica, lo referente a la máquina virtual de java (JVM), el JRE, entre otros aspectos.

Además se estudia lo referente a AJAX, dando sus beneficios, las ventajas que se tendrá al usar esta tecnología, sus desventajas, entre otros aspectos. Se dará una breve descripción de los IDE's que se puede usar en GWT lo cual va a servir para el siguiente capítulo de este trabajo investigativo.

Finalmente, se dará a conocer todo lo referente al estudio de GWT en donde se conocerá su instalación. Los requisitos y limitaciones que vamos a tener, la forma de comunicarse con el servidor usando los conocidos RPC (Llamadas a Procedimientos Remotos), se dará a

conocer todos los componentes para el desarrollo de GUI que se pueden dentro de GWT, así como su conexión con la base de datos.

2.2. Aspectos generales de Java

2.2.1. Lenguaje Java

2.2.1.1. Definición

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

Hoy en día, la tecnología Java ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma J2EE. Pero Java no se queda ahí, ya que en la industria para dispositivos móviles también hay una gran acogida para este lenguaje.

La tecnología Java está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. Con plataforma se refiere a la máquina virtual de Java (Java Virtual Machine).

Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que el programa escrito para Linux puede ser ejecutado en Windows sin ningún problema. Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas, entre otras.

Finalmente se concluye que Java brinda una solución para cada necesidad que pudiéramos tener.

2.2.1.2. Características del lenguaje

Java es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humano. También posee otras características muy importantes:

- Es un lenguaje que es compilado, generando ficheros de clases compilados, pero estas clases compiladas, son en realidad interpretadas por la máquina virtual de java. Siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando.
- Es un lenguaje multiplataforma: El mismo código java que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java.
- Es un lenguaje seguro: La máquina virtual, al ejecutar el código java, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.

Gracias al API de java se puede ampliar el lenguaje para que sea capaz, por ejemplo, comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas, crear aplicaciones visuales al estilo Windows.

Para poder trabajar con java es necesario emplear un software que permita desarrollar en java. Existen varias alternativas comerciales en el mercado: JBuilder, Visual Age, Visual Café, etc. y un conjunto de herramientas shareware, e incluso freeware, que permiten trabajar con java.

2.2.1.3. Máquina Virtual Java

Una Máquina Virtual Java (en inglés *Java Virtual Machine*, JVM) es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java *bytecode*), el cual es generado por el compilador del lenguaje Java.

El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable incluso como lenguaje de entrada para un microprocesador físico. Como todas las piezas del rompecabezas Java, fue desarrollado originalmente por Sun Microsystems.

La JVM es una de las piezas fundamentales de la plataforma Java. La gran ventaja de la máquina virtual java es aportar portabilidad al lenguaje de manera que desde Sun Microsystems se han creado diferentes máquinas virtuales java para diferentes arquitecturas

y así un programa .class escrito en un Windows que puede ser interpretado en un entorno Linux. Tan solo es necesario disponer de dicha máquina virtual para dichos entornos.

La máquina virtual de Java puede estar implementada en software, hardware, una herramienta de desarrollo o un Web browser.

2.2.1.4. JRE

El JRE (Java Runtime Environment, o Entorno en Tiempo de Ejecución de Java) es el software necesario para ejecutar cualquier aplicación desarrollada para la plataforma Java. El usuario final usa el JRE como parte de paquetes software o plugins (o conectores) en un navegador Web.

Sun ofrece también el SDK de Java 2, o JDK (Java Development Kit) en cuyo seno reside el JRE, e incluye herramientas como el compilador de Java, Javadoc para generar documentación o el depurador. Puede también obtenerse como un paquete independiente, y puede considerarse como el entorno necesario para ejecutar una aplicación Java, mientras que un desarrollador debe además contar con otras facilidades que ofrece el JDK.

2.2.1.5. JDK

El JDK es un súper conjunto de la JRE, y contiene todo lo que está en el JRE, además de herramientas tales como: los compiladores y depuradores necesarios para el desarrollo de applets y aplicaciones. El Java Runtime Environment (JRE) proporciona a las bibliotecas,

la máquina virtual Java, y otros componentes para ejecutar applets y aplicaciones escritas en el lenguaje de programación Java.

2.2.2. Aplicación web en Java

2.2.2.1. Aplicaciones web

En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una Intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información.

2.2.2.2. Aplicaciones web en Java

Para el desarrollo de aplicaciones web en Java se utilizan páginas web dinámicas, que son páginas que poseen funcionalidades especiales, para lo cual se utilizan otros lenguajes de programación, a parte del simple HTML. Mientras las paginas estáticas, todo el mundo se las puede imaginar, las páginas dinámicas son mucho más complejas a la vez que versátiles.

Las aplicaciones que se crean en grandes empresas deben ser más efectivas que eficientes, es decir, conseguir que el trabajo salga adelante y que la aplicación funcione, es más importante que la eficiencia con la que se realiza el trabajo, es por esto que al ser Java un lenguaje mucho más simple que cualquiera de los que se puede encontrar en la actualidad, permite concentrarse en la mecánica de aplicación que en otras tareas minuciosas.

Otro aspecto importante a destacar es la portabilidad de las aplicaciones en Java, ya que se pueden portar a cada una de las arquitecturas de las plataformas presentes en una empresa, lo que permite desarrollar aplicaciones corporativas para un grupo muy diferente de plataformas, lo que hará que los desarrolladores tiendan a realizar sus aplicaciones en lenguaje Java.

Las herramientas Java no presentan altos costos, al contrario, tienen precios de las herramientas de PCs, lo que hace que los programadores corporativos tengan un amplio campo donde moverse, muchas oportunidades y recursos a su disposición.

Estas páginas dinámicas son reconocidas, interpretadas y ejecutadas por el propio servidor, son útiles en muchas ocasiones, con ellas se puede hacer todo tipo de aplicaciones web, son especialmente útiles en trabajos en los que se tiene que acceder a información centralizada ubicada en una base de datos en el servidor y cuando por razones de seguridad los cálculos no se pueden realizar en el ordenador del usuario.

La ventaja de este tipo de programación es que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es HTML que es fácilmente interpretable.

El lenguaje Java explota más a fondo las características de las aplicaciones web, las que lo hacen que sea de gran utilidad para el desarrollo de aplicaciones web del lado del servidor, ya que su versatilidad permite realizar un espectro de tareas mucho más amplio, y de una manera mucho más a fondo, que se acoplan perfectamente a las necesidades de las aplicaciones web actuales.

2.3. Aspectos generales de AJAX

2.3.1. Introducción

Las aplicaciones AJAX son excelentes para la creación de aplicaciones web que son altamente interactivas y ofrecen un mejor servicio al usuario, mientras que es comparable a las aplicaciones de escritorio en funcionalidad, sin la molestia de descargar o instalar nada.

Las aplicaciones AJAX combinan los datos de intercambio XML, junto con HTML y CSS para el estilo de la interfaz, el objeto XMLHttpRequest para la comunicación asíncrona con el servidor de aplicaciones, y JavaScript para la interacción dinámica con los datos proporcionados. Esto le permite construir aplicaciones que forman parte de las aplicaciones Web 2.0 Web revolución, que rivalizan con una aplicación de escritorio en la respuesta. Se puede construir páginas web utilizando AJAX para comunicarse con el servidor en segundo plano, sin recargar la página. Incluso puede reemplazar a las distintas

secciones de la página web sin tener que actualizarse la página. AJAX, finalmente permite tomar las aplicaciones tradicionales de escritorio orientado, como procesadores de texto, hojas de cálculo y programas de dibujo, y servir a los usuarios a través de la Web.

2.3.2. Definición

AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, además es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dados que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Ajax es una combinación de cuatro tecnologías ya existentes:

- **XHTML** (o **HTML**) y hojas de estilos en cascada (**CSS**) para el diseño que acompaña a la información.

- Document Object Model (**DOM**) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto **XMLHttpRequest** para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto `iframe` en lugar del `XMLHttpRequest` para realizar dichos intercambios.
- **XML** es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente¹.

2.3.3. Antecedentes

A pesar de que el término "AJAX" fue creado en 2005, la historia de las tecnologías que permiten AJAX se remonta a una década antes con la iniciativa de Microsoft en el desarrollo de Scripting Remoto. Sin embargo, las técnicas para la carga asíncrona de contenidos en una página existente sin requerir recarga completa remontan al tiempo del elemento `iframe` (introducido en Internet Explorer 3 en 1996) y el tipo de elemento `layer` (introducido en Netscape 4 en 1997, abandonado durante las primeras etapas de desarrollo de Mozilla). Ambos tipos de elemento tenían el atributo `src` que podía tomar cualquier

¹ WIKIPEDIA La enciclopedia libre

dirección URL externa, y cargando una página que contenga javascript que manipule la página paterna, podían lograrse efectos parecidos al AJAX.

El **Microsoft's Remote Scripting** (o **MSRS**, introducido en 1998) realiza el envío de datos a través de un applet Java el cual se puede comunicar con el cliente usando JavaScript. Esta técnica funcionó en ambos navegadores, Internet Explorer versión 4 y Netscape Navigator versión 4. Microsoft la utilizó en el Outlook Web Access provisto con la versión 2000 de Microsoft Exchange Server.

Sin embargo, todavía se utilizan donde se requiere una mayor compatibilidad, una reducida implementación, o acceso cruzado entre sitios web. Una alternativa, el Terminal SVG (basado en SVG), emplea una conexión persistente para el intercambio continuo entre el navegador y el servidor.

2.3.3.1. OpenAjax Alliance

La OpenAjax Alliance es una organización de vendedores, proyectos de código abierto, y compañías usuarias de Ajax, dedicada a la exitosa adopción de tecnologías Web abiertas e interoperables basadas en Ajax. El principal objetivo es acelerar los éxitos de los usuarios con Ajax mediante la promoción de su habilidad para encontrar y combinar soluciones de proveedores de tecnología Ajax y ayudar a dirigir el futuro camino del ecosistema Ajax.

2.3.3.2. Modelo AJAX de Aplicaciones Web

En el gráfico II.1, II.2 y II.3 se aprecian las diferencias entre el modelo clásico de aplicación Web y el modelo Ajax de aplicación Web.

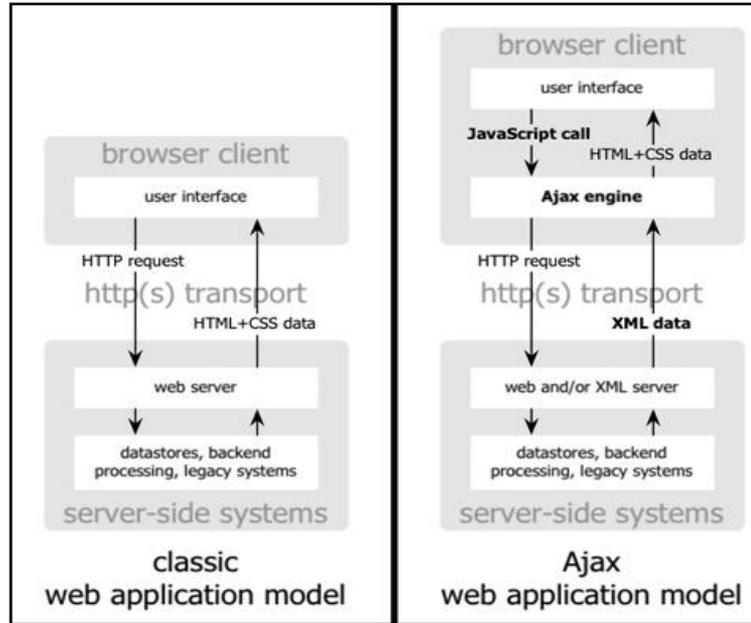


Gráfico II.1: Modelos comparativos de aplicaciones Web.

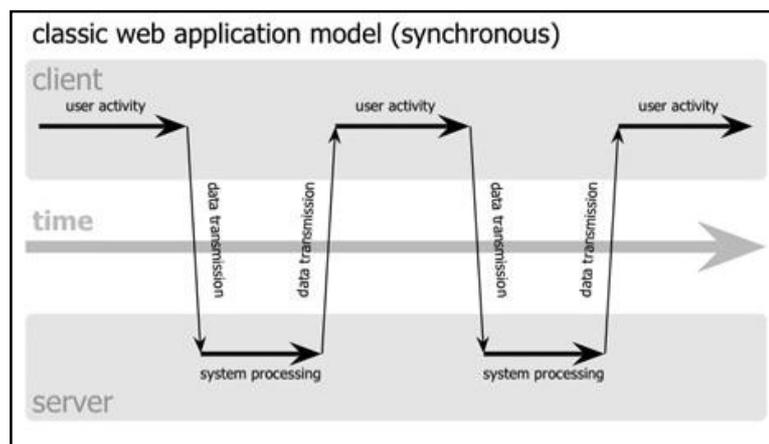


Gráfico II.2: Modelos clásico de aplicaciones Web.

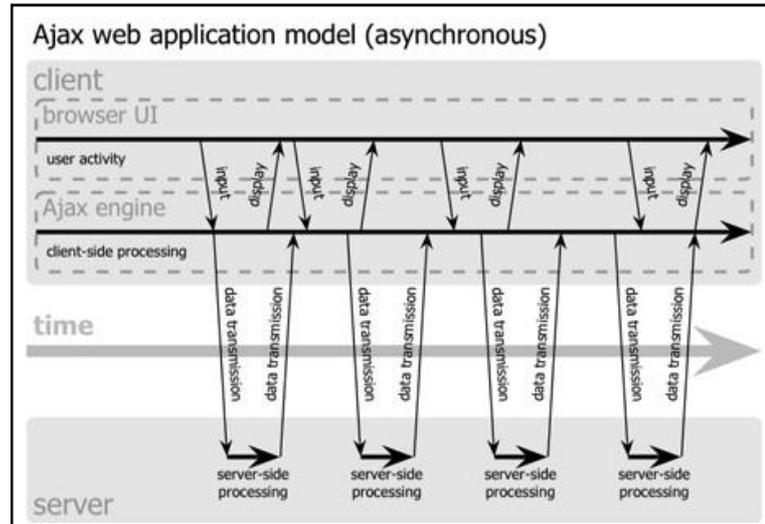


Gráfico II.3: Modelos AJAX de aplicaciones Web.

Las figuras mostradas precedentemente indican que la interacción entre el servidor y el navegador (cliente) no es directa en el modelo AJAX, sino que esta se realiza a través de los componentes AJAX, lo cual genera asincronismo y disminución en el tráfico entre el navegador y el servidor.

2.3.4. Importancia de AJAX

Ajax sirve principalmente para diseñar y programar interfaces de usuario mucho más allá de la web, rompiendo las limitaciones que la “sincronía” supone y abriendo una nueva puerta que nos permitirá desarrollar aplicaciones que en un principio solo podrían concebirse para el escritorio, en aplicaciones web. Este cambio permitirá implementar aplicaciones de escritorio con aplicaciones similares con el plus de la universalidad de Internet permitiendo acceder a información de manera remota pero siempre con una

interface similar y un numeroso grupo de utilidades que revalorizarán el producto convirtiendo cualquier desarrollo en algo más que una “página web”.

2.3.5. Beneficios

Existen 10 razones para utilizar Ajax como se describe a continuación:

- Basado en los estándares abiertos
- Usabilidad
- Válido en cualquier plataforma y navegador
- Beneficia las aplicaciones web
- No es difícil su utilización
- Compatible con Flash
- Adoptado por los “gordos” de la tecnología web
- Web 2.0
- Es independiente del tipo de tecnología de servidor que se utilice
- Mejora la estética de la web

2.3.6. Ventajas

- Permite diseñar interfaces muchísimo más dinámicas acercándonos a "aplicaciones de escritorio".
- La comunicación asíncrona con el servidor permite varias cosas que reducen el "peso de la página" / "líneas de código que el cliente tiene que descargarse".

- Campos de select, si las opciones son muchas... no es necesario transmitir las en la primera carga de la página, se puede producir la lista de opciones cuando el cliente pulse sobre el desplegable. Otra solución es hacer una búsqueda dinámica.
- La navegación por la aplicación mediante Ajax permite evitar al cliente descargar la cabecera de todos los "documentos", "pantallas" <html><head><tit..... etc... Se puede cambiar el contenido de cualquier objeto del DOM dinámicamente ante los eventos que controlamos con javascript.
- No requiere plugins o capacidades específicas de ciertos navegadores.
- Las aplicaciones son más interactivas, responden a las interacciones del usuario más rápidamente, al estilo desktop.
- Se reduce el tamaño de la información intercambiada.
 - Muchos micro-petición, pero el flujo de datos global es inferior.

2.3.7. Desventajas

- El cliente necesita un navegador que soporte javascript .Hoy por hoy la mayoría de los navegadores soporta JavaScript. Internet Explorer, Mozilla, Firefox, Safari etc. Para esta desventaja se tiene una "excusa", no se está hablando de desarrollar páginas web con Ajax, sino aplicaciones web, y como toda aplicación tiene unos requisitos mínimos, y siendo este javascript tampoco se discrimina un espectro muy amplio de usuarios.
- El mal uso de ajax/javascript puede mal emplearse sobrecargando el servidor de peticiones ej.: Si hacemos que cada milisegundo haga una consulta con una base de datos... al administrador del sistema le pueden surgir instintos asesinos.

2.4. ENTORNOS DE DESARROLLO INTEGRADOS (IDE's)

2.4.1. Tipos de IDEs

Entre los tipos de IDEs más conocidos tenemos los siguientes:

- GCJ – El compilador GNU para Java
- Eclipse
- NetBeans
- BlueJ
- Sun Java Studio
- JDeveloper
- JCreator

2.4.1.1. Descripción de IDEs

- **GCJ – El compilador GNU para Java**

GCJ es un compilador portable para Java. Este puede compilar código Java a Bytecodes (archivos .class), o directamente a código de máquina nativo, y de bytecode a código de máquina nativo.

Las aplicaciones compiladas son enlazadas con el Runtime de GCJ, libgcj, que contiene el núcleo de librerías de clases, un recolector de basura, y un intérprete de bytecode. libgcj puede cargar e interpretar dinámicamente archivos de clases, dando resultado a aplicaciones compiladas/interpretadas.

Gcj puede ser también configurado como un compilador multiplataforma, conveniente para sistemas de programación embebidos.

- **Eclipse**

Este es sin duda uno de los mejores (sino el mejor) IDE's de programación del mundo. Eclipse es un IDE multiplataforma libre para crear aplicaciones clientes de cualquier tipo.

El entorno integrado de desarrollo (IDE) de Eclipse emplea módulos (en inglés *plugin*) para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. El mecanismo de módulos permite que el entorno de desarrollo soporte otros lenguajes además de Java. Por ejemplo, existe un módulo para dar soporte a C/C++. Existen módulos para añadir un poco de todo, desde Telnet hasta soporte a bases de datos.

En cuanto a las aplicaciones clientes, Eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc. Por ejemplo, GEF (Graphic Editing Framework – Framework para la edición gráfica) es un plugin de Eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto WYSIWYG hasta editores de diagramas UML, interfaces gráficas para el usuario (GUI), etc. Dado que los editores realizados con GEF “viven” dentro de Eclipse, además de poder ser

usados conjuntamente con otros plugins, hacen uso de su interfaz gráfica personalizable y profesional.

- **NetBeans**

NetBeans es una herramienta para programadores que sirve para escribir, compilar, corregir errores y para ejecutar programas. Está escrito en Java – pero puede servir de soporte a cualquier otro lenguaje de programación. Existe también un número enorme de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de utilización.

- **BlueJ**

Este es un muy buen IDE de programación, y una de sus principales ventajas está en su simplicidad.

- **Sun Java Studio**

Sun One Studio es una plataforma gratis de desarrollo, especial para arquitecturas empresariales y desarrolladores que integran el centro de Desarrolladores de Red de Sun. Este incluye poderosas características para agilizar el desarrollo de aplicaciones, tales como un modelador UML, colaboración instantánea y perfiles de aplicación. El IDE también está totalmente integrado con un completo Runtime Environment: Sun Java Enterprise System.

- **JDeveloper**

Oracle JDeveloper es un IDE libre con end-to-end soporte para modelación, desarrollo, depuración, optimización, e implementación de aplicaciones Java y servicios Web.

- **JCreator**

JCreator es una herramienta de desarrollo para programadores de Java. Este IDE es rápido y sencillo, y lo recomiendo para personas que estén empezando ya que es muy intuitivo y fácil de manejar².

2.5. Plataforma RIA-GWT

2.5.1. Plataforma RIA (Rich Internet Application)

2.5.1.1. Historia

Los sistemas de software han estado alrededor por varias décadas, pero sólo hace poco tiempo que comenzó a ser utilizado por millones y millones de personas en todo el mundo. Hace sólo 20 años, la mayoría de las aplicaciones de software fueron utilizadas por profesionales entrenados.

El modelo de interacción es la que puede soportar una variedad de métodos de entrada y responder intuitivamente de manera oportuna. Como regla general, para ser considerado rica la interacción con el usuario es que debe ser tan buena como la actual generación de aplicaciones de escritorio, en lo referente a procesadores de texto y hojas de balance.

² Realizado por Cristian Castiblanco

Esto incluye características como la prestación de los diferentes medios de interacción por información visual por ejemplo, cambiando la forma del cursor, usando las indicaciones colores. En la siguiente figura se proporciona una visión general de las cuatro etapas en la evolución de las aplicaciones de software.

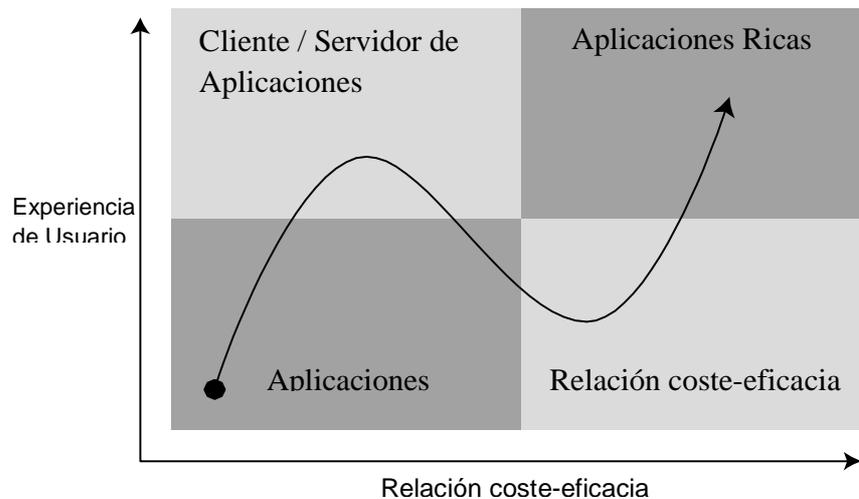


Grafico II.4: Visión General de la historia de las aplicaciones de software.

2.5.1.2. Ventajas de la Plataforma RIA

- **No requiere instalación.-** La aplicación es descargada y ejecutada automáticamente en el navegador. El software que corre actualmente la aplicación es instalada en la máquina del cliente.
- **Las actualizaciones son automáticas.-** las versiones nuevas de la aplicación también se descargarán automáticamente con sólo volver a visitar la página de la aplicación web.
- **Plataforma independiente.-** Una aplicación rica de Internet puede ejecutarse y operar potencialmente en cada plataforma de un navegador.

- **Más seguridad.-** Las aplicaciones se ejecutan en un ambiente restringido en el navegador y por lo tanto son menos probables que sea dañada como las aplicaciones que necesitan ser instalados.
- **Más respuesta.-** Porque no todas las acciones de los usuarios requieren comunicación con el servidor, las RIA tienden a ser más sensibles que las aplicaciones web clásicas.
- **Más escalable.-** Una gran parte del trabajo computacional, así como también el cuidado estatal puede ser descargado desde el cliente, por lo que el servidor puede manejar muchos más usuarios.
- **Más eficiencia de red.-** En las aplicaciones web clásicas, cada acción del usuario requiere que el servidor cargue toda la página y enviarla a la red. En el caso de las aplicaciones ricas en Internet (RIA), toda la aplicación de interfaz de usuario sólo tiene que ser comunicado una vez y todas las otras peticiones al servidor requieren sólo los datos actuales que se envían al cliente³.

2.5.1.3. Desventajas de la Plataforma RIA

- **Requiere JavaScript o un plug-in específico.-** Porque toda la aplicación se ejecuta en el cliente a través del intérprete de JavaScript. Cuando el usuario desactiva JavaScript por completo la aplicación por lo general hace poco o nada. Obviamente es posible tener un plan alternativo para esos usuarios, pero luego se va a mantener dos aplicaciones separadas, que está lejos de ser ideal.

³ Por: Bram Smeets, Uri Boness y Ronald Bankras

- **No hay acceso a los recursos del sistema.-** Como las aplicaciones AJAX se ejecutan dentro de un navegador, ellos están limitados en los recursos que pueden acceder. Por ejemplo, una aplicación AJAX no puede tener acceso al sistema de archivos del cliente.
- **Es duro de indexar completamente para los motores de búsqueda.-** Porque la mayoría de los motores de búsqueda no soportan que aplicaciones actualicen parcialmente las páginas o usen un plug-in específico tales como Flash. La mayoría de las aplicaciones dinámicas de Internet están mal indexadas por los motores de búsqueda.
- **Cuestiones de accesibilidad.-** Al hacer actualizaciones parciales de página mediante JavaScript o un plugin específico puede romper la accesibilidad. El problema más grande y más notoria es que no se pueda manejar correctamente la pantalla.
- **Depende de una conexión a Internet.-** Porque estas aplicaciones son servidas desde la web y ejecutadas en el navegador web, por lo que requieren por lo menos una conexión a Internet inicial. Pero incluso durante su uso, es necesaria una conexión a Internet para comunicarse con el servidor. Cuando la conexión no está temporalmente disponible, las RIA's no funcionará⁴.

2.5.2. Framework GWT

2.5.2.1. Introducción

Google Web Toolkit (GWT) es un framework de código abierto desarrollado por Google y lanzado en mayo del 2006 bajo la licencia Apache 2.0. con el objetivo de facilitar el

⁴ Por: Bram Smeets, Uri Boness y Ronald Bankras

desarrollo de Aplicaciones Ricas de Internet (RIAs), que permite escapar de la “matriz” de tecnologías usadas actualmente para escribir aplicaciones AJAX, las cuales son difíciles de manejar y propensas a errores.

El corazón de GWT es un compilador que compila código JavaScript a partir de código Java. El compilador es capaz de generar código multibrowser, aliviando la tarea del desarrollador de escribirlo por sí mismo. GWT genera código interpretable por los browser más importantes del mercado (Mozilla Firefox, Safari, Opera e Internet Explorer). Cabe aclarar que este compilador, a diferencia del framework de desarrollo, **no es Open Source**.

GWT proporciona un entorno de desarrollo basado en Java que permite construir aplicaciones AJAX utilizando dicho lenguaje. El procedimiento consiste en que primero se encapsula el objeto XMLHttpRequest API, para luego minimizar los problemas entre exploradores. De esta manera, se puede construir rápido y eficiente aplicaciones AJAX sin preocuparse demasiado acerca de la adaptación de su código para que funcione en varios navegadores.

Además, permite aprovechar el Standard Widget Toolkit (SWT) o Swing estilo de programación, que proporciona un marco que permite combinar los widgets en interfaces de usuario. Esta es una gran manera de mejorar la productividad y acortar sus líneas de tiempo de desarrollo, aprovechando el conocimiento del lenguaje de programación Java y la familiaridad con el marco de interfaz de desarrollo basado en eventos.

GWT proporciona además un conjunto de widgets listos para utilizar en la interfaz de usuario que se puede utilizar de inmediato para crear nuevas aplicaciones. También proporciona una forma sencilla de crear widgets innovadores mediante la combinación de las ya existentes. Para crear, depurar y probar cada unidad de aplicaciones AJAX se puede utilizar el IDE de Eclipse. Así mismo, se puede construir RPC (Remote Procedure Call) llamadas a procedimientos remotos, que son servicios que proporcionan ciertas funcionalidades para acceder de forma asíncrona a la base de datos y a sus aplicaciones web fácilmente utilizando el marco de GWT RPC.

GWT permite integrar fácilmente con los servidores descritos en otros idiomas, así se puede mejorar las aplicaciones para proporcionar una experiencia de usuario mucho mejor mediante la utilización del Framework AJAX.

2.5.2.2. Definiciones de GWT

- GWT (Google Web Toolkit) es un framework Open Source, desarrollado por Google, con el objetivo de facilitar el desarrollo de Aplicaciones Ricas de Internet (RIAs).
- El corazón de GWT es un compilador que compila código JavaScript y código HTML a partir de código Java. Además, es capaz de generar código multi-browser, aliviando la tarea del desarrollador de escribirlo por sí mismo. GWT genera código interpretable por los browser más importantes del mercado (Mozilla Firefox, Safari, Opera e Internet Explorer). Cabe aclarar que este compilador, a diferencia del framework de desarrollo, **no es Open Source** al momento de trabajar con algún diseñador gráfico de interfaces.

- GWT permite que el desarrollador escriba todo el FrontEnd de un sistema web, simplemente en Java, olvidándose de programar JSPs, JavaScripts, HTML, etc. El manejo de varios lenguajes al mismo tiempo, lleva a código poco confiable, difícil de mantener y casi imposible de debuggear.
- Una de las tecnologías más importantes en los últimos años en el mundo de las aplicaciones web es AJAX. Esta tecnología, comúnmente difícil de integrar en aplicaciones desarrolladas en grandes frameworks como Struts o Java Server Faces (JSF), es la *esencia* de las aplicaciones ricas de internet, y como GWT está pensado para desarrollar RIAs, la *esencia* de GWT es entonces AJAX.
- AJAX es inmensamente sencillo de *Utilizar* en una aplicación desarrollada con GWT. Y no solamente sencillo desarrollar, es sencillo de *Mantener*, y es sencillo de *Debuggear*⁵.

2.5.2.3. Características de GWT

El compilador GWT es justamente un mecanismo para entregar enteramente una aplicación arquitectura cliente/servidor. Las características incluyen:

- **Componentes de la interfaz de usuario dinámicos y reutilizables.**- Un sistema de UI widgets estándar que parece bueno, flexible, que se contempla para trabajar en todos los navegadores importantes (incluyendo safari y opera).
- **RPC realmente fácil.**- Para comunicarte desde el navegador que lanza la aplicación con el servidor web, solamente se necesita definir clases de Java serializables para las

⁵ Realizado por Epiwiki deploying ideas

peticiones y respuestas. En producción, GWT serializa automáticamente las peticiones del navegador y de-serializa las respuestas desde el servidor web. El mecanismo de RPC de GWT puede incluso manejar jerarquía de polimorfismo en clases, y puede manejar las posibles excepciones.

- **Administración del historial del navegador.-** Las aplicaciones en AJAX no necesitan utilizar el botón “Atrás” (o Back) del navegador. Y GWT no es la excepción, es decir, no es necesario que se llame a otras páginas para realizar las diferentes acciones, ni recargar el navegador.
- **Depuración en tiempo real.-** Para cuando la aplicación esté lista, el código de la misma es traducida a JavaScript, pero mientras lo estás desarrollando este corre sobre una máquina virtual de Java (JVM). Lo que significa que en la fase de desarrollo se tiene la posibilidad de depurar la aplicación con los avanzados sistemas de debugging y manipulación de excepciones incluidos en IDEs como Eclipse.
- **Compatibilidad con los navegadores.-** Las aplicaciones en GWT serán automáticamente soportadas por navegadores como FireFox, Internet Explorer, Mozilla, Safari, y Opera sin ningún tipo de operación para la detección de los mismos, en la mayoría de los casos.
- **Integración con JUnit.-** Mediante la integración de JUnit en GWT, se puede probar las aplicaciones y depurarlas en un navegador mientras las construyes. Se puede testear llamadas asíncronas a procedimientos remotos RPC.
- **Internacionalización.-** Se crea aplicaciones y librerías de Internacionalización rápida y fácilmente.

- **Interoperabilidad y control.-** Si las librerías de clases de GWT no son suficientes para lo que se necesita, se puede mezclar JavaScript en el código de la aplicación usando la interfaz nativa de scripts de Java (JavaScript Native Interface, JSNI).
- **GWT es un proyecto de código abierto.-** Todo el código de GWT está disponible bajo la licencia Apache 2.0 ⁶.
- Soporte para las API's de Google (inicialmente, soporte para Google Gears).
- Se crea aplicaciones Ajax y aún escribir los códigos en lenguaje Java.
- Se puede usar los entornos de desarrollos integrados (IDE) de siempre.
- Se puede eliminar errores del cliente en ese IDE java.
- Comunicación entre el cliente y servidor usando objetos de java y es todo mucho más ligero en el cliente de lo que sería en un java applet.
- Es de código abierto.
- Los desarrolladores pueden diseñar y desarrollar sus aplicaciones orientadas a objetos. Errores comunes en Javascript, como la discrepancia de tipos de datos, son controlados en tiempo de compilación.
- El código Javascript generado puede ser ofuscado para optimizar el rendimiento.
- Existe un numeroso conjunto de bibliotecas desarrolladas por Google y terceros que amplían las funcionalidades de GWT.
- Depuración en tiempo real.
- Ayuda para eliminar errores completamente equipados de Java.
- GWT maneja todas las ediciones del cruz-browser para el revelador.

⁶ WIKIPEDIA La enciclopedia libre

2.5.2.4. Arquitectura GWT

GWT tiene cuatro componentes principales: un compilador Java-a-JavaScript, un navegador web "hosted", y dos librerías de clases. A continuación presentaremos estos componentes en tres diferentes figuras, para de esta manera alcanzar un mejor entendimiento.

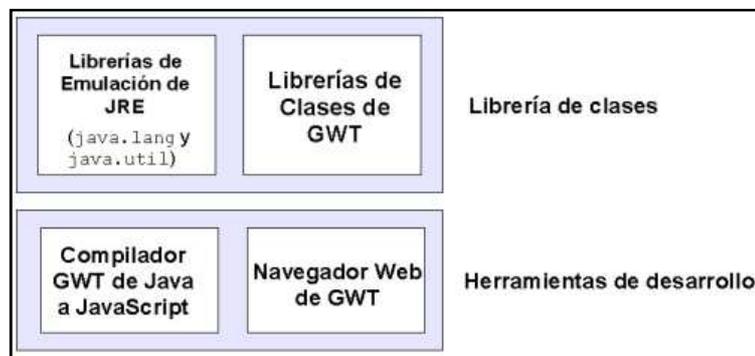


Gráfico II.5: Arquitectura de GWT.

Los componentes son:

- **Compilador GWT Java-a-JavaScript**

El Compilador GWT Java-a-JavaScript traduce del lenguaje de programación Java a JavaScript. Tiene tres modos en los que puede operar, los cuales determinan cómo se verá:

- *Obfuscate*: por defecto. Genera código inentendible y altamente optimizado.
- *Pretty*: genera código JavaScript entendible.
- *Detailed*: produce código al igual que el modo pretty pero agregando el nombre completo de la clase como parte del nombre de las funciones.

El código JavaScript es generado en archivos para cada tipo de navegador e idioma. Al iniciar el navegador el mismo ejecuta código que permite obtener la versión correcta del archivo para la aplicación.

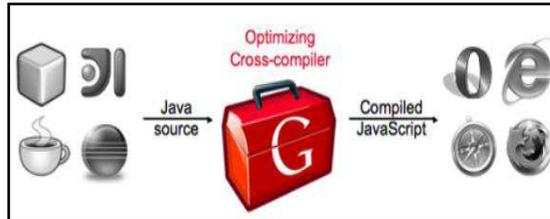


Gráfico II.6: Compilador GWT Java-a-JavaScript

- **Navegador web “Hosted” de GWT**

El Navegador web “Hosted” de GWT permite correr y ejecutar aplicaciones GWT en modo hosted, donde lo que se está corriendo son bytecodes de Java sobre una máquina virtual sin compilarlos a JavaScript. Para lograr esto, el navegador GWT incrusta un controlador de browser especial (un control del Internet Explorer sobre Windows o un control de Gecko/Mozilla sobre Linux) con hooks dentro de la máquina virtual de Java.

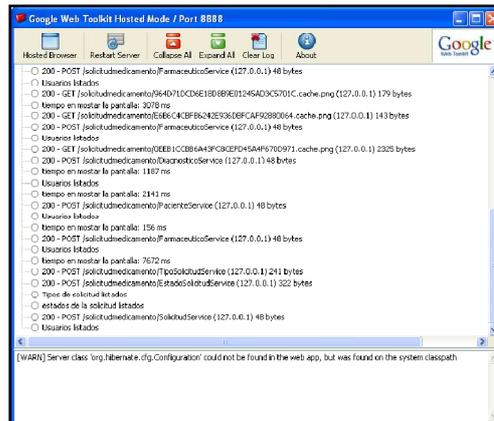


Gráfico II.7: Navegador Web “Hosted” de GWT

- **Emulación de librerías JRE**

Google Web Toolkit soporta solamente una pequeña parte de las librerías de clases disponibles en Java 2 Standard Edition y Enterprise Edition, esto sucede ya que éstas librerías son bastante grandes y usan funcionalidades que no están disponibles en navegadores web. Para saber completamente qué clases y métodos son soportados por el núcleo de paquetes Java, se debe ver la referencia del API para *java.lang* y *java.util*, en donde se listan las clases a las que se les dio soporte y contiene notas acerca de las diferencias entre lo soportado y el estándar de Java. Por ejemplo, las clases de los paquetes como *java.io* no se utilizan en aplicaciones web ya que estas acceden a recursos en la red y al sistema de archivos local.

Algunas áreas específicas en las que la emulación Google Web Toolkit difiere desde el estándar de Java son:

- **Expresiones regulares**

La sintaxis para expresiones regulares en Java es similar, pero no idéntico, a las expresiones regulares en JavaScript. Por ejemplo, los métodos `replaceAll` y `split` usan expresiones regulares. Así que, se debe ser cuidadoso al usar expresiones regulares que tienen el mismo significado tanto en Java como en JavaScript.

- **Serialización**

La serialización Java requiere de unos cuantos mecanismos que no están disponibles en JavaScript, como por ejemplo el cargar dinámicamente clases y la reflexión. Por consiguiente, Google Web Toolkit no soporta el estándar de serialización de Java. En lugar de eso, Google Web Toolkit tiene una facilidad de soporte para RPC, que

proporciona serialización automática de objetos desde y hacia el servidor usando invocación de métodos remotos.

- **Librería de clases de interfaz de usuario de GWT**

Las librerías de clases de interfaz de usuario de GWT son un conjunto de interfaces y clases personalizadas que permiten crear "widgets" para el navegador, como botones, cajas de texto, imágenes, etc. Éste es el núcleo de las librerías de interfaz de usuario para crear aplicaciones GWT ⁷.



Gráfico II.8: Librería de clases de interfaz de usuario de GWT

2.5.2.5. Para qué sirve el framework GWT

Lo que hace especial a este framework es que las aplicaciones se programan en Java, y el compilador GWT se encarga de generar a partir de las fuentes el código JavaScript. El framework que permite la utilización de librerías JavaScript directamente desde Java mediante JSNI.

⁷ Por: Cristian Castiblanco

El mecanismo que ofrece GWT para comunicarse con el servidor está basado en RPC que aunque funciona y hace que el código cliente sea sencillo de leer carece de las ventajas de otros tipos de arquitecturas.

Si se quiere acceder a funcionalidad en el servidor haciendo peticiones HTTP en vez de llamadas RPC se puede usar la clase RequestBuilder, una clase de bajo nivel cuyo uso requiere tener conocimientos HTTP.

2.5.2.6. Funciones y Funcionamiento de GWT

2.5.2.6.1. Funciones

- **Comunicarse con el servidor mediante una sencillísima llamada RPC.-** GWT admite un conjunto indefinido de protocolos de transferencia, como JSON y XML, pero el mecanismo de llamada a procedimiento remoto (RPC) de GWT permite el establecimiento de comunicaciones Java de una forma especialmente sencilla y eficaz. Al realizar una llamada a un método remoto desde el navegador, el mecanismo RPC de GWT serializará automáticamente los argumentos, ejecutará el método adecuado en el servidor y anulará la serialización del valor de retorno del código cliente.
- **Utiliza componentes de la interfaz de usuario en varios proyectos.-** Se puede crear artilugios reutilizables mediante la composición de otros artilugios y colocarlos después fácilmente en paneles. Para reutilizar el artilugio en otro proyecto, sólo se tiene que prepararlo para que se pueda usar en un archivo JAR.

- **Utiliza otras bibliotecas JavaScript y código JavaScript original.-** Si la biblioteca de clases de GWT no satisface las necesidades, se puede mezclar manualmente JavaScript en el código fuente Java mediante la interfaz JSNI. Por tanto, la ventaja de poder crear modelos de objetos JS como tipos de Java (por ejemplo, creación de código, reestructuración, sustitución de llamadas a funciones, etc.) sin tener que contar con una mayor cantidad de memoria ni sufrir una disminución de velocidad. Esta función posibilita un uso óptimo de las estructuras JSON.
- **Fácil uso del historial y del botón de retroceso del navegador.-** las aplicaciones AJAX no necesitan desgastar el botón de retroceso del navegador. GWT hace que resulte más sencillo utilizar el sitio con sólo añadir el estado al historial del botón de retroceso del navegador.
- **Localiza las aplicaciones de una forma eficaz.-** Puede crear fácilmente bibliotecas y aplicaciones internacionalizadas con las potentes técnicas de vinculación aplazada de GWT. Además, a partir de la versión 1.5, los artilugios GWT estándar admiten la bidireccionalidad.
- **Elige las herramientas de desarrollo de forma productiva.-** GWT utiliza Java, por lo que se puede utilizar todas las herramientas favoritas de desarrollo con Java (Eclipse, IntelliJ, JProfiler, JUnit, NetBeans, etc.) al crear las aplicaciones AJAX. De esa forma, el desarrollador web, beneficiará el aumento de productividad derivado de la creación/solicitud de código y de la reestructuración de Java automatizada.
- **Prueba el código con JUnit.-** La integración directa de GWT con JUnit permite efectuar la comprobación por unidad tanto en un depurador como en un navegador, e

incluso comprobar por unidad las llamadas a procedimientos remotos (RPC) asíncronas.

- **Extensión o contribución con el software libre Google Web Toolkit.-** Todo el código de GWT está disponible bajo la licencia de Apache 2.0 ⁸.

2.5.2.6.2. Funcionamiento

Google Web Toolkit (GWT) permite crear aplicaciones AJAX en el lenguaje de programación Java que son compiladas posteriormente por GWT en código JavaScript ejecutable optimizado que funciona automáticamente en los principales navegadores.

Durante el desarrollo de una aplicación, puede repetir rápidamente el mismo ciclo "editar - actualizar - ver" típico de JavaScript y aprovechar la ventaja añadida de poder depurar y recorrer una a una todas las líneas de código Java. Cuando ya se está listo para la implementación, GWT compilará el código fuente Java en archivos JavaScript optimizados independientes. Google Web Toolkit permite crear fácilmente tanto un artilugio para una página web como una aplicación completa⁸.

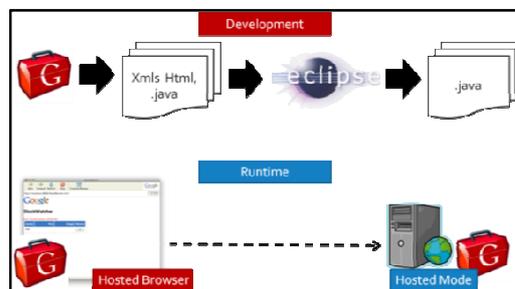


Gráfico II.9: Funcionamiento de GWT

⁸ Por: Compañía Google

2.5.2.7. Ventajas de GWT

- Es similar a una aplicación de escritorio, con un alto dinamismo de las pantallas.
- No necesita conocimientos Javascript.
- Desarrollo ágil.
- Permite avanzar desde un prototipo.
- Multiplataforma, Multinavegador.
- Crea modelos de objetos JS como tipos de Java (por ejemplo, creación de código, reestructuración, sustitución de llamadas a funciones, etc.) sin tener que contar con una mayor cantidad de memoria ni sufrir una disminución de velocidad.
- Reduce el ancho de banda una vez cargada la aplicación en la caché.
- Reduce la carga en el servidor (los datos que se cargaban en la sesión del servidor ahora pasan al cliente).
- Permite la misma seguridad que otros frameworks (inclusive es más complicado inyectar datos debido al sistema de serialización que utiliza).
- Los errores comunes JavaScripts (types, type mismatches) son capturados por el programador durante compilación en vez de por los usuarios durante la ejecución ⁹.

2.5.2.8. Desventajas de GWT

- Solo desarrolladores JAVA.
- Curva de aprendizaje lenta al principio si el programador tiene mucha experiencia en otra tecnología.

- Dependiendo de las características de la aplicación puede ser un poco más costosa el despliegue.
- Consumo de memoria en el navegador.
- No es 100% API Java Objects
 - `Java.lang.java.util`
 - Reflection, Synchronized estas librerías no usa GWT.
- *Compilación.*- Demora mucho tiempo al momento que se compila desde el navegador propio de GWT a cualquiera de los browser que se desee presentar o ejecutar ⁹.

2.5.2.9. Ciclo de desarrollo

El ciclo de Desarrollo de GWT sería el siguiente:

- a. Usa el entorno de desarrollo integrado (IDE) favorito para el desarrollador para escribir y depurar una aplicación en Java, usando las librerías GWT que necesita.
- b. Usa el compilador de Java a JavaScript de GWT para transformar la aplicación en un conjunto de archivos JavaScript y HTML que puede colgar en cualquier servidor y ejecutar desde un navegador web.
- c. Verifica que las aplicaciones trabajen sobre todos y cada uno de los navegadores que consideras que los clientes usarán.

En GWT se puede usar componentes de interfaz de usuario llamados Widgets, para construir aplicaciones AJAX con GUIs (interfaz de usuario gráfico) atractivas. Al igual que

⁹Realizado por: De Benedetti Guido Rubén

en la mayoría de los lenguajes de programación, los componentes de la UI (interfaz de usuario) son agrupados en paneles que determinan la ubicación de los mismos. A continuación veamos una completa aplicación que utiliza un botón y un manejador de eventos:

```
public class Hola implements EntryPoint{
    public void onModuleLoad(){
        Button b = new Button("Click aquí", new ClickListener(){
            public void onClick(Widget sender){
                Window.alert("Hola,");
            } });
        RootPanel.get().add(b);
    } }
```

2.5.2.10. Modelo de programación en GWT

Lo que más impacta del modelo de programación que propone GWT es que se elimina la programación en JavaScript. Eso no significa que lo que se vaya a ejecutar en el navegador no vaya a ser JavaScript y HTML. GWT cambia drásticamente la forma en que hasta ahora se han desarrollado las aplicaciones web y propone que toda la aplicación (tanto cliente, como servidor) se programe usando un solo lenguaje, Java.

Esto no significa que GWT proponga Applets para las interfaces de usuario. Este nuevo framework se encarga automáticamente de "compilar" la parte cliente de la aplicación a DHTML y JavaScript. De esta manera, se puede usar las potentes herramientas de desarrollo del mundo Java (Eclipse, NetBeans, IntelliJ, etc.).

El compilador GWT es capaz de compilar (por el momento) sólo un subconjunto del API disponible en Java aunque suele ser más que suficiente para el tipo de aplicaciones que se va a desarrollar. Para las clases cliente, se usa los propios paquetes de GWT (com.google.gwt.user.client.*).

Otra característica interesante es que GWT ofrece dos modos distintos de ejecución. El denominado "modo web" o "web mode" consiste en compilar el código con el compilador GWT y ejecutarlo en un navegador habitual. Este modo es el que se usará cuando la aplicación se ejecute en producción pero seguimos sin poder depurar fácilmente la aplicación durante su desarrollo. Para ello GWT propone un segundo modo de ejecución denominado "hosted mode", que consiste en ejecutar la aplicación dentro de una JVM.

En este caso el código Java no es compilado a JavaScript y por tanto se puede ejecutar la aplicación paso a paso, tal y como depuraríamos cualquier aplicación en un IDE favorito. Para utilizar este modo de ejecución, dentro del jar de GWT (gwt-user.jar) se incluye una aplicación (llamada shell o consola) que es capaz de ejecutar tanto el código cliente (en un navegador propio) como el código servidor (en un pequeño motor de servlets basado en Tomcat).

Para quien haya programado interfaces de usuario en Java, ya sea mediante AWT, Swing o SWT, programar con GWT le va a resultar muy fácil. GWT incluye una amplia colección de controles (widgets) que incluye desde los más básicos (Labels, Button, TextBox, etc)

hasta algunos más complejos y no disponibles en HTML de forma directa (MenuBar, Tree o StackPanel).

2.5.2.11. Estructura de un proyecto GWT

Un proyecto GWT, pese a contener aplicaciones web, tiene una estructura más similar a un proyecto Java que a un proyecto J2EE (estructura WAR). GWT necesita ciertos ficheros adicionales y el seguimiento de una cierta nomenclatura en los paquetes Java. El primer ejemplo se lo va a llamar "Principal" y se utilizará el nombre de paquete "com.ree.ree". La tabla 1 muestra la estructura que debería tener este proyecto.

Tabla II.1. Estructura de proyecto

src/
src/com/ree/ree/Principal.gwt.xml
src/com/ree/ree/client/
src/com/ree/ree/client/Principal.java
src/com/ree/ree/server/ (opcional)
src/com/ree/ree/public/
src/com/ree/ree/public/Principal.html
bin/
www/

La carpeta "src" contiene:

- Las fuentes Java para el lado cliente que serán compilados a JavaScript. Carpeta "
src/com/ree/ree/client/

- Las fuentes de los servicios RCP del servidor (que se compilarán a .class). Carpeta "`src/com/ree/ree/server/`".
- Los recursos estáticos que necesite la aplicación (imágenes, css, etc.). Carpeta "`src/com/ree/ree/public/`".
- El descriptor de módulo (`src/com/ree/reePrincipal.gwt.xml`). GWT basa sus aplicaciones en módulos, y cada módulo debe tener su fichero de configuración. El concepto es similar al de descriptor de despliegue en J2EE. En este fichero se declaran los servicios RPC que van a ser invocados desde el cliente así como los punto de entrada a nuestra aplicación (entrypoints).
- Página contenedora (`src/com/ree/ree/public/Principal.html`). Aunque se ha dicho que con GWT no es necesario escribir código HTML, existe una pequeña excepción a esta regla. Debe existir al menos una página HTML que contenga nuestra aplicación. Se puede por ejemplo crear un diseño HTML que muestre una serie de banners publicitarios con un espacio central que será donde se ubique la aplicación. Esta es la página que deberá invocarse desde un navegador para lanzar la aplicación GWT.

La carpeta "bin" contiene el resultado de compilar las clases Java mediante el compilador javac (es decir, ficheros .class). Hay que notar que aquí se compilarán todas las clases Java, incluso las del lado cliente. Esto es así, ya que cuando ejecutemos la aplicación en el "hosted Mode" se ejecutarán estas clases y no el resultado de la compilación a JavaScript.

La carpeta "www" contiene la aplicación lista para desplegar en un servidor web. Es decir, aquí se encuentran las clases cliente, compiladas a JavaScript mediante el compilador

GWT. El compilador también se encarga de copiar aquí el contenido de la carpeta `src/com.ree.ree/public` que antes se ha mencionado. Para probar la aplicación basta con abrir el fichero `www/com.ree.ree/Principal.html` en un navegador y se estará ejecutando la aplicación en el modo web (web mode). El contenido de esta carpeta se puede desplegar en un simple servidor web ya que todo es HTML, JavaScript y XML.

Cabe recalcar que GWT incluye una utilidad (`applicationCreator.cmd`) que la encargada de crear todas estas carpetas es por el usuario. Además de la estructura del proyecto anteriormente mencionada, se crearán dos scripts:

- `Principal-compile.cmd`. Este script será necesario invocar al compilador GWT y generar el código cliente en la carpeta "www".
- `Principal-shell.cmd`. Script que permitirá ejecutar la aplicación en "hosted mode" mientras estemos desarrollando la aplicación.

El fichero generado "`src\com\ree\ree\client\Principal.java`", inicialmente contiene código para un simple HelloWorld, por lo que ya se puede lanzar estos dos scripts y ver sus resultados.

El script de compilación (`Principal-compile.cmd`) creará la carpeta "www" con nuestra aplicación cliente ya compilada. En este punto se puede abrir en nuestro navegador la página "`www\com.ree.ree.Principal\Principal.html`" y veremos la aplicación en ejecución en modo web.

Si por el contrario se lanza el script para la ejecución en "hosted mode": (Principal-shell.cmd) se lanzará la consola de GWT y un navegador que automáticamente apuntará a la página contenedora "Principal.html".

La clase principal implementa la interfaz "com.google.gwt.core.client.EntryPoint". Esta interfaz obliga a implementar el método "onModuleLoad()" que se ejecuta cuando una página HTML declara el uso de un módulo GWT. Por último, la clase "RootPanel" nos permite acceder a los elementos HTML que se ha definido en la página "Principal.html".

Además de proporcionar los "contenedores" donde podremos ubicar los controles que necesite la aplicación, la página "Principal.html" debe declarar el uso del módulo que se quiere utilizar. Eso se hace dentro de la cabecera () mediante la etiqueta:

```
<meta name='gwt:module' content='com.ree.ree.Principal'/>
```

También es necesario incluir la referencia al fichero JavaScript "gwt.js" que se encarga de ejecutar el código del módulo(s) que declare la página. Este fichero JS ha sido creado por la utilidad applicationCreator y no es necesario que se modifique su contenido:

```
<script language="javascript" src="com.ree.ree.Principal.nocache.js"></script>
```

Por lo demás, se es libre de añadir el código HTML que se desee. Normalmente se suele añadir una referencia a una hoja de estilos para personalizar la apariencia de los controles gráficos de GWT.

2.5.2.12. Los módulos en GWT

Un módulo es un *paquete* de fácil distribución que contiene código del lado del cliente en la aplicación, implementación de servicios en el lado del servidor y recursos varios, como archivos estilos o librerías JavaScript. Un módulo se configura en los archivos llamados *Modulo.gwt.xml*. En estos archivos se configura el punto de entrada del módulo, la ruta al directorio donde se encuentra el código compilable por el compilador, y los servicios asincrónicos que estarán disponibles, entre otras cosas. Un módulo puede heredar de otro módulo, e *importar* de esta manera todos los recursos que el módulo padre ofrece.

Ejemplo del modulo.gwt.xml

```
<module>
  <inherits name="com.google.gwt.user.User"/>
  <entry-point class="com.ree.ree.client.Nombre_Proyecto"/>
</module>
```

El módulo hereda del módulo *User* de GWT, e *importa* de esta manera todos los componentes de aquel módulo. Las unidades individuales de configuraciones en GWT son archivos XML llamados módulos.

Un módulo reúne todos los datos de configuración que el proyecto GWT necesita, es decir:

- Módulos heredados.
- Un nombre de clase; esto es opcional, aunque cualquier módulo referido a un HTML debe tener al menos una clase entry-point especificada.

- Entradas a los source paths
- Entradas a los public paths

Los módulos pueden aparecer en cualquier paquete del classpath, aunque es altamente recomendable que estos aparezcan en el paquete raíz de un proyecto estándar.

Clases entry-point

Un módulo **entry-point** es cualquier clase que es asignable a `EntryPoint` y que puede ser construida sin parámetros. Cuando un módulo es cargado, cada clase entry point es instanciada y el método `EntryPoint.onModuleLoad()` es llamado.

Source Path

Los módulos pueden especificar qué subpaquetes contienen código fuente traducible, provocando que el paquete nombrado y sus subpaquetes sean añadidos al source path. Solamente los archivos encontrados en el *source path* son candidatos para ser traducidos a JavaScript, haciendo posible que se mezclen códigos fuentes del lado del cliente (client-side) con los del lado del servidor (server-side) en el mismo classpath sin ningún tipo de conflicto.

Cuando un módulo hereda de otro, sus *source path* son combinados así que cada módulo tendrá acceso al código fuente traducible que requiera.

Public path

Los módulos pueden especificar qué subpaquetes son públicos, provocando que el paquete nombrado y sus subpaquetes sean añadidos al public path. Cuando se compila la aplicación a JavaScript, todos los archivos que pueden ser encontrados sobre el *public path* son copiados al directorio de salida de los módulos. El efecto en la red es que las URLs visibles al usuario no necesitan incluir un nombre de paquete completo.

Cuando un módulo hereda de otro módulo, sus public paths son combinados así que cada módulo tendrá acceso al recurso estático que requiera.

Especificaciones

- ***Formato de módulos XML.***- Los módulos son definidos en XML y situados dentro de la jerarquía de paquetes de tu proyecto
- ***Inclusión Automática de paquetes.***- Los módulos contienen referencias a archivos JavaScript y CSS externos, causando que estos sean cargados cuando el módulo mismo es cargado.
- ***Filtrado de paquetes públicos.***- Filtra archivos dentro y fuera de tu *public path* para evitar la publicación accidental de archivos.

2.5.2.12.1. Formatos de los módulos XML

Los módulos son definidos en ficheros XML cuya extensión de archivo es *.gwt.xml*. Los archivos de módulos XML deben residir en el paquete raíz del proyecto.

Usando una estructura estándar de proyecto, el módulo XML puede ser tan simple como esto:

```
<module>
  <inherits name="com.google.gwt.user.User"/>
  <entry-point class="com.example.cal.client.CalendarApp"/>
</module>
```

Cargando módulos

Los archivos de módulos XML son encontrados en el classpath de Java, referenciados por su propio nombre de módulo desde las páginas alojadas en el proyecto, y por ser heredados por otros módulos.

Los módulos son siempre referenciados por sus nombres lógicos. El nombre lógico de un módulo es del tipo pkg1.pkg2.NombreModulo (aunque pueden haber cualquier número de paquetes) y excluyendo incluso la extensión del archivo. Por ejemplo, el nombre lógico de un archivo de módulo XML situado en:

```
~/src/com/ree/ree/Principal.gwt.xml
es
com.ree.ree.Principal
```

Elementos disponibles

- *<inherits name="logical-module-name"/>*.- Hereda todas las configuraciones desde el módulo especificado como si el contenido del módulo XML heredado fuera copiado literalmente.

- `<entry-point class="classname"/>`.- Especifica una clase *entry point*.
- `<source path="path"/>`.- Añade paquetes al *source path* para combinar los paquetes en que el módulo XML es encontrado con el path especificado al subpaquete.
- `<public path="path"/>`.- Añade paquetes al *public path* para combinar el paquete en el que el módulo XML es encontrado con el path especificado para identificar la raíz del *public path*.
- `<servlet path="url-path" class="classname"/>`.- Para el uso conveniente de RPC, este elemento carga una clase servlet montada como el path URL especificado. El path URL debe ser absoluto y tener la forma de un directorio (por ejemplo, /spellchek). El código cliente entonces, especifica éste mapeo de URL en una llamada al método.

ServiceDefTarget.setServiceEntryPoint(String).

- `<script src="js-url"/>`.- Inyecta automáticamente archivos JavaScript externos localizados en la ruta *src*.
- `<stylesheet src="css-url"/>`.- Inyecta automáticamente hojas de estilo en cascada situadas en archivos externos, localizados en la ruta *src*.
- `<extend-property name="client-property-name" values="comma-separated-values"/>`.- Extiende el conjunto de valores para una propiedad del cliente existente. Cualquier número de valores puede ser añadido de ésta manera, y los valores de la propiedad del cliente acumulado desde módulos heredados. Esto es de mucha importancia si se va a crear un proyecto con Internacionalización.

2.5.2.12.2. Inclusión automática de recursos

Los módulos pueden contener referencias a archivos JavaScript y CSS externos, causando que estos sean cargados automáticamente cuando el módulo mismo es cargado.

Incluyendo JavaScript Externo

La inclusión de scripts es una forma conveniente de asociar archivos JavaScript externos con nuestro módulo de manera automática. La siguiente sintaxis se usa para cargar un archivo JavaScript externo dentro de la página alojada en nuestro proyecto, antes de que el entry point de nuestro módulo sea llamado:

```
<script src="js-url"/>
```

El script es cargado dentro del namespace de la página como si lo hubieras incluido explícitamente usando la etiqueta `<script>` de HTML. El script será cargado antes que el método `onLoadModule()` sea llamado.

Incluyendo Hojas de Estilo Externas

La inclusión de hojas de estilo es una excelente forma de asociar archivos CSS externos con nuestro módulo de una manera automática. La siguiente sintaxis se usa para añadir archivos CSS externos a la página de nuestro módulo:

```
<stylesheet src="css-url"/>
```

Se puede añadir cualquier número de hojas de estilo de ésta manera, y el orden de la inclusión dentro de la página refleja el orden en que estos elementos aparecen en nuestro módulo XML.

Inclusión y herencia de módulos

La herencia de módulos crea recursos de inclusión particularmente convenientes. Si se desea crear una librería re-utilizable que depende de archivos JavaScript y CSS particulares, se puede estar seguro que los clientes de nuestra librería tienen todo lo que se necesita automáticamente, usando herencia desde nuestro módulo.

2.5.2.12.3. Filtrando paquetes públicos.

Los elementos `<public>` soportan ciertos atributos y elementos anidados para permitir patrones basados en inclusión y exclusión. Esto utiliza las mismas reglas que un elemento `FileSet` de Ant.

El elemento `<public>` no soporta completamente la semántica de `FileSet`. Solamente los siguientes atributos y elementos anidados son soportados:

- El atributo `include`
- El atributo `exclude`
- El atributo `defaultexcludes`
- El atributo `casesensitive`
- Etiquetas `include` anidadas
- Etiquetas `exclude` anidadas

Los demás atributos y elementos anidados no son soportados.

2.5.2.13. Herramientas de la línea de comandos

GWT te ofrece un pequeño set de herramientas de línea de comandos fáciles de manejar, para realizar diferentes tareas de una manera rápida. Éstas son también útiles para añadir nuevas cosas a los proyectos existentes. Por ejemplo, `projectCreator` podría ser usado para crear un proyecto Eclipse para uno de los ejemplos que viene con GWT.

2.5.2.13.1. `projectCreator`

El programa **`projectCreator`** es el encargado de crear una configuración de proyecto apropiada, por ejemplo, para Eclipse.

```
projectCreator -eclipse Nombre_Proyecto -out Nombre_Directorio
```

2.5.2.13.2. `applicationCreator`

El programa **`applicationCreator`** es el encargado de crear una estructura de aplicación GWT. Es programa crea los llamados *Módulos* de GWT

```
applicationCreator -eclipse Nombre_Proyecto -out Nombre_Directorio  
com.ree.ree.client.Nombre_Proyecto
```

Con el comando previo, se crea sobre el proyecto *Nombre_Proyecto*, en el directorio *Nombre_Directorio*, un módulo llamado *Nombre_Proyecto*, cuyo punto de entrada es la clase *com.ree.ree.client.Nombre_Proyecto*.

2.5.2.13.3. junitCreator

Genera un test JUnit.

2.5.2.13.4. i18nCreator

Genera un archivo de propiedades i18n y un script de sincronización.

2.5.2.14. Construcción de Interfaces de Usuarios GUI

Las clases de interfaces de usuario de Google Web Toolkit son muy similares a las de los distintos frameworks como Swing o SWT, excepto que los widgets son creados y renderizados usando HTML creado dinámicamente.

Mientras sea posible manipular el DOM del navegador directamente usando la interfaz DOM, es más fácil usar las clases desde la jerarquía de widgets. Raramente se necesitará usar DOM directamente. Usando widgets puede construir interfaces de usuario rápidamente, y que se comporten de manera adecuada sobre todos los navegadores.

La clase RootPanel

La clase **RootPanel** es una abstracción para acceder al cuerpo de la página host, transparentemente desde Java. Esta clase representa el elemento **body** del documento HTML. Todos los widgets serán agregados directa o indirectamente al **RootPanel** para ser vistos por el usuario.

Métodos para acceder al panel raíz:

- `RootPanel.get()`: Devuelve el panel raíz por defecto (el elemento **body**)
- `RootPanel.get(id)`: Devuelve el panel raíz identificado por el id dado.

2.5.2.14.1. Widget y paneles

WIDGETS

Los widget son los componentes y partes visuales de una aplicación GWT los cuales son vistos por los usuarios que usan la página web, como por ejemplo labels, botones, imágenes, menús, grids etc. La GUI de una aplicación GWT está basada en Widgets.

Una definición formal de widget puede ser la siguiente:

“Los widgets son los componentes visibles de una aplicación GWT que un usuario puede ver en la página por la que esta navegando.”

Una vez que se tiene una buena idea de lo que es un widget , se dará una mirada a los widgets que vienen con GWT, incluida la forma de usarlos en los diferentes componentes que nos proporciona GWT.

Definición de un Widget

El paquete `com.google.gwt.user.client.ui` contiene las clases que permiten crear interfaces de usuario dinámicas usando técnicas que ya se han probado exitosamente con otros frameworks que proveen esta funcionalidad como por ejemplo el paquete AWT de Java. Las clases en este paquete aprovechan las características de las interfaces de usuario de un

navegador web para proveer componentes dinámicos reutilizables, los cuales tienen el mismo comportamiento en las versiones e implementaciones de los navegadores soportados.

La librería de interface de usuario de GWT llama a estos componentes widgets, y estos van desde simples botones o labels hasta controles más complejos como tabs o arboles. Algunos de estos widgets se traducen directamente en elementos HTML que normalmente se usarían en la elaboración de una página web, mientras que otros son la composición de muchos elementos HTML combinados con scripts y manejo de eventos.

Los Widgets son uno de los cuatro bloques fundamentales de construcción de aplicaciones GWT (los otros son los panels, eventos y la comunicación con el servidor).

Cuando un usuario está utilizando una aplicación GWT hecha por alguien, lo que se encuentra mirando es un grupo de Widgets distribuidos de cierta manera en paneles y que responderán a eventos. Los widgets son los componentes con los que el usuario interactúa.

GWT proporciona muchos widgets diferentes, entre estos los controles más usados en las interfaces graficas como son los botones, labels, menús cuadros de textos, etc.



Gráfico II.10: Ejemplo de un widget (botón)

Muchas de nuestras aplicaciones serán construidas usando varios widgets los cuales se distribuirán de cierta manera dentro de paneles para darle alguna estructura a la interfaz de la aplicación.

Usando widgets como objetos Java

El propósito general de GWT es desarrollar Rich Internet Applications en Java y después el compilador de GWT genera el código HTML y java script necesario para que la aplicación funcione bien en los diferentes navegadores existentes. Para eso se hace necesaria una manera de representar varios objetos del navegador, es decir los llamados widgets de GWT, en nuestro código Java.

Se tomará ventaja de la habilidad de la programación orientada a objetos para modelar objetos y conceptos como objetos de programación. Por ejemplo, en una aplicación GWT cualquiera se hace uso de un objeto Java llamado Button este objeto Button modelara varias propiedades que esperamos que un botón tenga, como la habilidad de asignarle un texto además de presionar dicho botón. Así como un botón se puede modelar los demás componentes que se desee ver en la aplicación, es decir los widgets, como objetos Java con métodos y propiedades.

Entonces, en una programación diaria en GWT se debe considerar todos los widgets como objetos naturales de Java. El botón del que se habla anteriormente se crea llamando al constructor del Button de GWT como sigue:

```
Button boton = new Button("Aceptar");
```

Este código crea un nuevo objeto Button de GWT del cual después se puede usar varios métodos.

Considerando Widgets como elementos DOM

La representación Java de los widgets funciona muy bien en el código java y permite construir una aplicación GWT como se quiera, usando cualquier cantidad de widgets y sus métodos asociados para construir la funcionalidad de la aplicación. Sin embargo no se puede mostrar esos objetos Java en el navegador, si no se tiene todavía una aplicación AJAX.

No se debe confiar en el hecho de que un widget en particular es implementado como un elemento DOM en particular, ya que dicho widget puede ser implementado usando una representación DOM diferente en versiones futuras. Si no se concentra en usar el código java usando los métodos que nos da cada widget de GWT se protegerá contra cambios futuros a nivel DOM.

Widgets Estándar

La versión estándar de GWT viene con un amplio número de widgets para usar en nuestras aplicaciones. Estos widgets cubren las áreas que se espera con botones, cuadros de texto y otros widgets. Sin embargo habrá ocasiones en las que se quiere usar algún widget, como una barra de progreso, pero que no viene dentro de los widgets que GWT proporciona pero que se podrá crear.

Dentro del conjunto de widgets, los diseñadores de GWT han implementado una fuerte jerarquía de clases de Java con el fin de proporcionar un elemento de coherencia en los widgets donde dicha coherencia naturalmente exista. Por ejemplo los widgets `TextBox`, `TextArea` y `PasswordTextBox`, comparten ciertas propiedades. GWT reconoce esto y captura las propiedades comunes en una clase llamada `TextBoxBase` de la cual heredan estos 3 widgets. Para ilustrar las jerarquías se muestra en el siguiente gráfico.

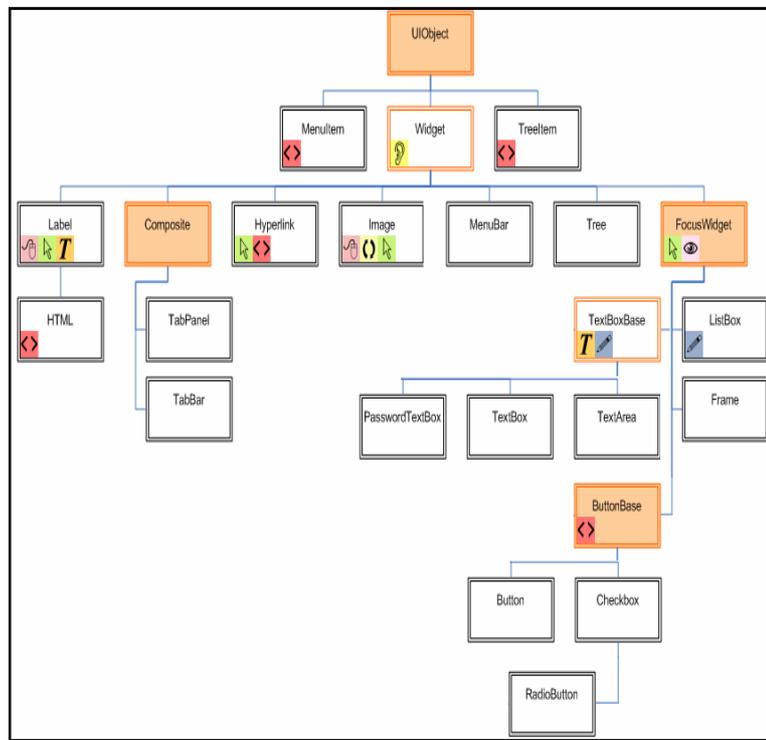


Gráfico II.11: Jerarquía de clases de los widgets GWT.

Se puede ver en la jerarquía de clases que todos los widgets heredan de la clase `UIObject`, la cual proporciona un conjunto de métodos y atributos para todos los widgets, incluyendo el tamaño, visibilidad y nombres de estilos. Es dentro de esta clase que encontramos el método `setElement()`.

La clase `UIObject` permite el acceso a una variedad de funcionalidades DOM sin que se tenga acceso directo al DOM. Por ejemplo es posible asignar el `height` (alto) de un `UIObject` usando el método `setHeight()`, el cual usa el método `setStyleAttribute()` de la clase `DOM`.

```
public void setHeight(String height) {  
    DOM.setStyleAttribute(element, "height", height);  
}
```

Después de `UIObject` todos los widgets, excepto `TreeItem` y `MenuItem`, deben heredar de la clase `Widget`, la cual proporciona todas las funcionalidades a los widgets, incluyendo métodos que son llamados cuando un widget es agregado o eliminado de un panel. Esta clase también contiene la implementación por defecto del método `onBrowserEvent`, el cual permite a un widget manejar cualquier evento que tenga relacionado.

A continuación se muestra los widgets que nos proporciona gratis GWT, así como la manera de usarlos en nuestra aplicación, además se dividirá los diferentes tipos de widgets en las siguientes categorías:

- **Widgets estáticos**, son aquellos que no son muy interactivos, solo cambian de estado como resultado de alguna orden dada por la aplicación.
- **Widgets de formularios**, encontraremos aquellos elementos que normalmente son usados para formularios, incluyendo botones y cuadros de texto.

- **Widgets complejos**, los cuales son nuevas características, brindadas por las interfaces de usuario, para navegadores, son composiciones de etiquetas HTML y manejo de eventos con java script.

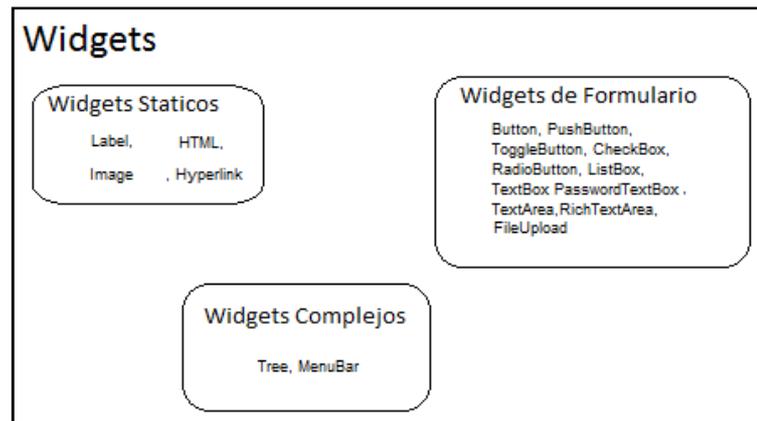


Gráfico II.12: Categorías de los Widgets

Widgets Estáticos

Estos widgets no tienen ningún estado interno ni tampoco cambian dinámicamente por su cuenta. Sin embargo pueden ser parte de una interface de usuario dinámica en la cual sus propiedades pueden ser cambiadas en tiempo de ejecución por medio del código de otros controles, pero sus propiedades no cambian como resultado de acciones del usuario. Estos widgets incluyen: Label, HTML, Image y el Hyperlink.

Label

Un Label contiene un texto específico, el cual es visualizado exactamente como es escrito. Esto significa que el Label creado por el código `new Label ("hola Mundo")`

aparecerá en el navegador exactamente así "Hola Mundo", la palabra Mundo no es interpretada como HTML y no será visualizada en negrilla.

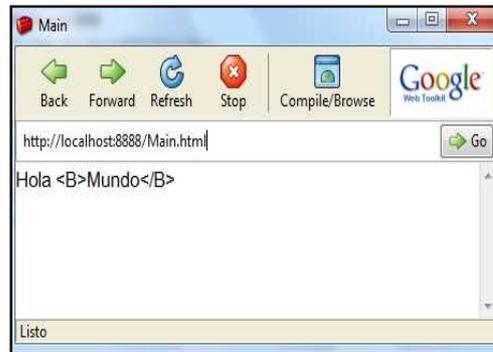


Gráfico II.13: Apariencia del Widget Label

Un Label tiene asociado por defecto un nombre de clase CSS: gwt-Label, el cual permite fácilmente asignar el estilo para todos los Label de nuestra aplicación. Por ejemplo, agregando el siguiente código CSS hará que el Label tenga un fondo azul con la fuente amarilla:

```
.gwt-Label { background-color:#008; color:#FF0;}
```

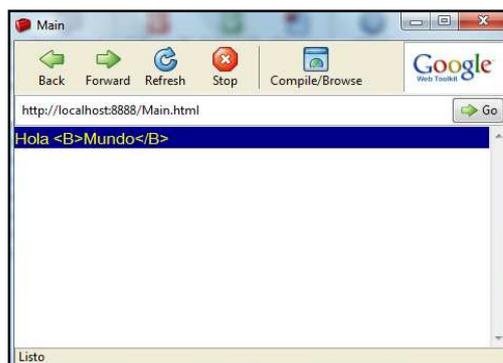


Gráfico II.14: Apariencia de un label al aplicarle css

HTML

El Widget HTML es muy similar al Label pero con la pequeña diferencia que puede soportar y renderizar HTML. De hecho este widget hereda de Label, ganando su soporte para eventos del mouse y alineación de texto. Este widget interpreta cualquier texto como HTML. Si en un Label el texto “Hola Mundo” se vería tal y cual se escribió, en un HTML se vería “Hola **Mundo**”.



Gráfico II.15: Apariencia del Widget HTML

Su estilo de CSS predefinido es gwt-html, esto permite personalizar la apariencia a través de hojas de estilo así como el Label. Este control tiene algunas desventajas, ya que puede generar problemas de seguridad u otros debido a que puede llevar código malicioso.

Image

El widget Image es estático igual que el Label y el HTML, pero en vez de aceptar y renderizar una cadena, el Image acepta una URL apuntando a un archivo de imagen y renderiza dicha imagen. El widget básicamente lo que permite es utilizar la etiqueta img de HTML como un widget del framework. El siguiente código permite crear una imagen:

```
Image imagen = new Image();  
imagen.setUrl("archivo.jpg");  
RootPanel.get().add(imagen);
```



Gráfico II.16: Apariencia del Widget Image

Lo que se hace es crear una instancia de la clase “image”, luego se pasa la url de la imagen que se desea mostrar a través del método “setUrl”, y finalmente se agrega al panel principal. El método “setUrl” acepta una cadena con la url donde se encuentra la imagen, dicha url puede ser una dirección local como en el ejemplo que se vio donde el navegador busca una imagen que se llama “archivo.jpg” en el mismo directorio del HTML.

Pero también se puede colocar la url de la imagen en otro servidor, por ejemplo si ahora se usa esta url “http://www.google.com.co/intl/en_com/images/logo_plain.png“, muestra lo siguiente.



Gráfico II.17: Apariencia del Widget Image con URL externa

Hyperlink

Este widget actúa como un hipervínculo dentro de nuestra aplicación GWT. Para el usuario aparenta no ser más que un vínculo normal dentro de la página web el cual al dar clic se realizará una navegación dentro de la página. En el código esta acción lo que se hace es manipular el objeto “History” de GWT para cambiar el estado de la aplicación.

Cualquier componente que use este widget debe implementar la interface **ValueChangeHandler**, que es la que maneja los eventos relacionados con cambios de estados en la aplicación, e implementar el método **onValueChange**, que es el que se ejecuta cuando se realiza un cambio en algún estado de la aplicación, en el cual irá las acciones a tomar dentro de la aplicación cada vez que un Hyperlink es utilizado. Además hay que agregar la siguiente línea al HTML de la aplicación:

```
<iframe id="gwt_historyFrame" style="width:0;height:0;border:0"></iframe>
```

Para utilizar un Hyperlink se crea una instancia de la clase Hyperlink y luego se agrega el widget a un panel.

```
Hyperlink h1 = new Hyperlink("Hyperlink 1","imagen1");
```

El constructor del Hyperlink recibe como parámetros 2 Strings, el primero el texto del link, y el segundo una cadena que se convierte en el estado del objeto “History” que se mencionó anteriormente, al dar clic en un Hyperlink esta cadena se concatena junto con un carácter ‘#’ a la url de la aplicación. A continuación se muestra un ejemplo:

```
public class Main implements EntryPoint, ValueChangeHandler<String>{
    Image imagen;
    public void onModuleLoad() {
        History.addValueChangeHandler(this);
        imagen = new Image();
        imagen.setUrl("http://fluxqubit.files.wordpress.com/2008/03/0074.jpg");

        Hyperlink h1 = new Hyperlink("Imagen 1","imagen1");
        Hyperlink h2 = new Hyperlink("Imagen 2","imagen2");

        RootPanel.get().add(imagen);
        RootPanel.get().add(h1);
        RootPanel.get().add(h2);
    }

    @Override
    public void onValueChange(ValueChangeEvent<String> event) {
        String historia = (event.getValue());

        if (historia.equals("imagen1")){
            //RootPanel.get().add(l);
            imagen.setUrl("http://fluxqubit.files.wordpress.com/2008/03/0074.jpg");
        }
        else if(historia.equals("imagen2")){
            //l.removeFromParent();
            imagen.setUrl("http://blogs.jetbrains.com/pti/images/GWT.png");
        }
    }
}
```

Gráfico II.18: Ejemplo del código de uso del Hyperlink

En la aplicación del ejemplo lo que se hace es agregar 2 Hyperlinks llamados h1 y h2 los cuales al hacer clic en cada uno de ellos cambian una imagen que se muestra en el browser.

La aplicación se vería algo así:



Gráfico II.19: Ejemplo gráfico del uso del Hyperlink

Con los Hyperlinks se puede agregar controles dinámicamente a nuestra aplicación, eliminar paneles o widgets, en fin permite tener una experiencia de navegación dentro del sitio web. Otra buena forma de navegación dentro de nuestra aplicación sería utilizando menús. Los Hyperlinks funcionan muy bien para navegar dentro de la aplicación, pero si lo que se quiere son links hacia otros sitios sería mejor utilizar un link HTML usando el widget HTML que ya se vio.

Widgets de Formulario

Los que se llama “Widgets de formulario” son aquellos que son usados normalmente en formularios HTML. Sin embargo los formularios HTML envían información al servidor y luego se muestra el resultado refrescando la página. Las aplicaciones AJAX intentan

eliminar el refresco de la página y en cambio envían y reciben información del servidor de manera asíncrona.

Estos “Widgets de formulario” proporcionados con GWT no necesitan ser incluidos dentro de un formulario HTML son muy flexibles y pueden ser usados de manera muy similar a widgets de aplicaciones de escritorio. Los widgets que se verá a continuación son los siguientes:

Button ToggleButton, PushButton, RichTextArea, CheckBox, RadioButton, ListBox, TextBox, PasswordTextBox, TextArea y FileUpload.

Button, ToggleButton y PushButton

El widget Button se comporta como los botones clásicos de cualquier aplicación de escritorio o web, es el widget GWT equivalente input de tipo “Button” de un formulario HTML. El funcionamiento de este widget es igual al de cualquier botón, normalmente los botones se usan en un formulario HTML para enviar información al servidor, pero en GWT se puede usar un botón para ejecutar cualquier acción que necesitemos. Para crear un botón se debe hacer algo como lo que sigue:

```
Button boton = new Button(“Click”)
```

A continuación se muestra un ejemplo en el cual utilizamos un botón.

```
8 public class FormWidgets implements EntryPoint {
9
10     public void onModuleLoad() {
11         // TODO Auto-generated method stub
12         Button boton = new Button("Click");
13         boton.addClickHandler(new ClickHandler() {
14             @Override
15             public void onClick(ClickEvent event) {
16                 // TODO Auto-generated method stub
17                 RootPanel.get().add(new Label("Hola Mundo"));
18             }});
19
20         RootPanel.get().add(boton);
21     }
22 }
```

Gráfico II.20: Creación del Widget Button

En el ejemplo anterior se creó un botón y lo agregó al panel principal, a dicho botón se le agregó un manejador del evento click el cual se definió con una clase anónima. El método `onClick` es el que se ejecutará al invocar al evento click, lo que hace el ejemplo es mostrar un “Hola Mundo” cada vez que se haga clic en el botón.

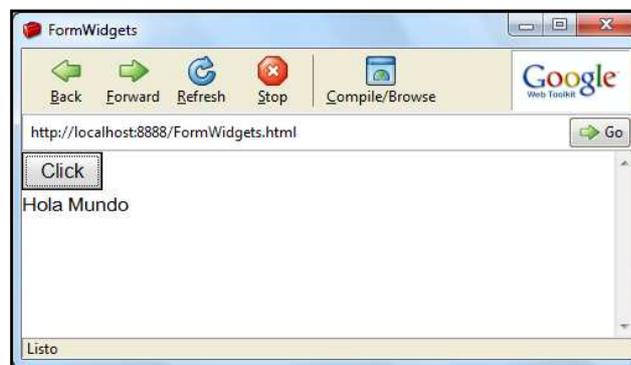


Gráfico II.21: Modo gráfico del widget Button

El `ToggleButton` y `PushButton` son otros 2 tipos de botones que vienen dentro de la librería de la interface de usuario que pueden ser usados de manera similar a un widget `Button` normal. Un ***ToggleButton*** se diferencia de un `Button` en que cuando se hace clic en estos se mantienen presionados hasta que el usuario hace clic de nuevo, además tiene varios

constructores los cuales aceptan una imagen como parámetro para reemplazar la cara del botón. Los eventos de clic y demás es igual a como se definió los del Button anteriormente. A continuación se muestra el uso de un ToggleButton.

```
8 public class FormWidgets implements EntryPoint , ClickHandler {
9
10     Label lbl = new Label("Hola");
11     public void onModuleLoad() {
12
13         ToggleButton boton = new ToggleButton(new Image("mas.png"), new Image("menos.png"));
14         boton.addClickHandler(this);
15         RootPanel.get().add(boton);
16     }
17     @Override
18     public void onClick(ClickEvent event) {
19
20         if(((ToggleButton)event.getSource()).isDown()){
21             RootPanel.get().add(lbl);
22         }
23         else{
24             lbl.removeFromParent();
25         }
26     }
27 }
28
```

Gráfico II.22: Creación del Widget ToggleButton

El **PushButton** también puede recibir una imagen en el constructor como el ToggleButton, además soporta algo de personalización en cuanto a estilos CSS.

El PushButton y ToggleButton comparten varios estados que pueden ser personalizables desde una hoja de estilo CSS los cuales no son soportados por un simple Button. Estos estados son seis y se describe a continuación.

- *Button up*: es el estado normal de un PushButton o ToggleButton.
- *Button down*: el estado cuando el clic del mouse está presionando un PushButton o el estado down de un ToggleButton.

- *Button up mouse hovering*: es la misma que el Button up con la diferencia que el mouse en esos momentos se encuentra sobre el PushButton o ToggleButton.
- *Button up disable*: igual que el Button up pero el botón en ese momento se encuentra deshabilitado por el método setDisable del botón.
- *Button down mouse hovering*: mismo estado que el Button down pero el mouse se encuentra sobre el botón en esos momentos.
- *Button down disable*: este estado nada más funciona para el ToggleButton ya que es el único que puede estar deshabilitado y en el estado down al mismo tiempo.

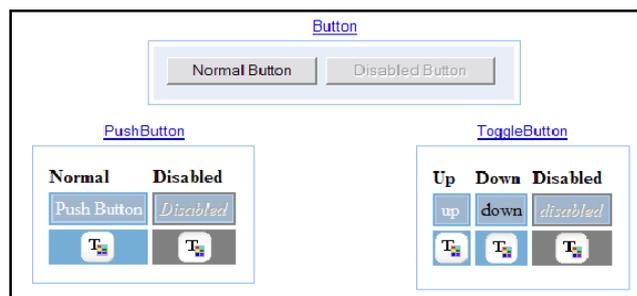


Gráfico II.23: Estado de los botones.

CheckBox

El widget CheckBox es la implementación del CheckBox estándar de HTML. Este widget soporta eventos de cambio de foco y eventos de clic. Cuenta con un estado que dice si el widget está chequeado o no, que es la función `getValue` la cual nos diría *true* si se encuentra chequeado y *false* obviamente lo contrario. A continuación se muestra un ejemplo del uso del CheckBox haciendo una lista de acciones pendientes.

```
8 public class FormWidgets implements EntryPoint, ClickHandler {
9
10 public void onLoad() {
11
12     VerticalPanel lista = new VerticalPanel();
13
14     CheckBox item1 = new CheckBox("Sacar la basura");
15     item1.addClickHandler(this);
16
17     CheckBox item2 = new CheckBox("Sacar al perro");
18     item2.addClickHandler(this);
19
20     CheckBox item3 = new CheckBox("Lavar la ropa");
21     item3.addClickHandler(this);
22
23     CheckBox item4 = new CheckBox("Comprar la comida");
24     item4.addClickHandler(this);
25
26     lista.add(item1);
27     lista.add(item2);
28     lista.add(item3);
29     lista.add(item4);
30
31     RootPanel.get().add(lista);
32 }
33 @Override
34 public void onClick(ClickEvent event) {
35
36     if(((CheckBox)event.getSource()).getValue()){
37         ((CheckBox)event.getSource()).setStyleName("gwt-CheckBox-checked");
38     }
39     else{
40         ((CheckBox)event.getSource()).setStyleName("gwt-CheckBox");
41     }
42 }
43 }
```

Gráfico II.24: Creación de un widget CheckBox

Lo que se hace en este ejemplo es simplemente definir 4 CheckBox y agregarlos a un VerticalPanel para que los CheckBox se agreguen uno debajo del otro.

RadioButton

Este widget es igual de sencillo y parecido al CheckBox que se acabó de ver, con la diferencia de que varios RadioButton asociados al mismo grupo no pueden estar

seleccionados a la vez. Este widget es útil cuando necesitamos que el usuario elija un solo ítem entre un grupo de la misma categoría. A continuación se muestra un ejemplo:

```
8 public class FormWidgets implements EntryPoint {
9
10 public void onModuleLoad() {
11
12     RadioButton item1 = new RadioButton("Colores", "Azul");
13     RadioButton item2 = new RadioButton("Colores", "Verde");
14     RadioButton item3 = new RadioButton("Colores", "Rojo");
15
16     RootPanel.get().add(item1);
17     RootPanel.get().add(item2);
18     RootPanel.get().add(item3);
19 }
20
21 }
```

Gráfico II.25: Creación de un RadioButton

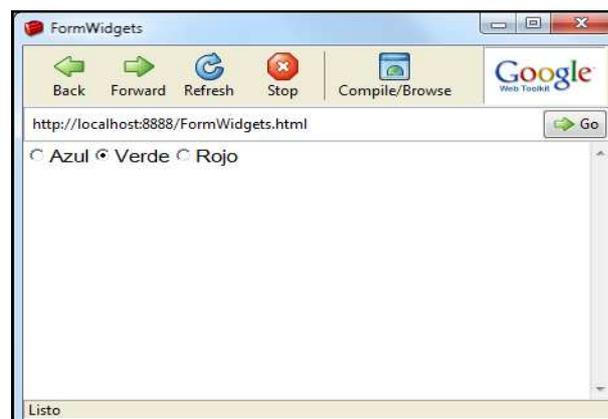


Gráfico II.26: Modo gráfico del widget RadioButton

ListBox

Un ListBox es un widget parecido al RadioButton, es decir muestra al usuario varias opciones y lo pone a elegir una sola dentro de las múltiples opciones. La diferencia con el RadioButton es la forma de mostrar las opciones, como se muestra a continuación:

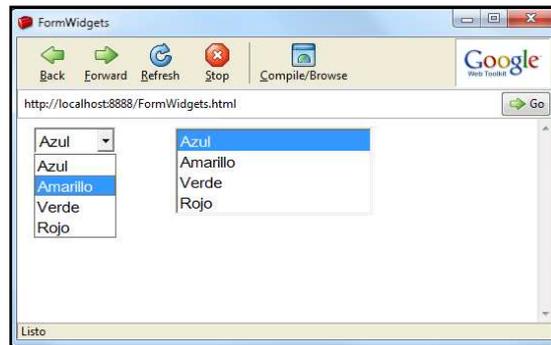


Gráfico II.27: Modo gráfico del widget ListBox

La primera forma muestra al ListBox como una lista desplegable que muestra solo la opción seleccionada, cuando el usuario hace clic en la flecha se despliega las demás opciones como se ve en el gráfico anterior. La forma de crear este ListBox es la siguiente:

```
ListBox lb = new ListBox();  
lb.addItem("Azul");  
lb.addItem("Amarillo");  
lb.addItem("Verde");  
lb.addItem("Rojo");  
  
RootPanel.get().add(lb);
```

Gráfico II.28: Creación de un ListBox (primera manera)

La segunda forma para mostrar un ListBox consiste en una lista de las opciones, donde la opción que se encuentra seleccionada se encuentra resaltada. La forma de definir un ListBox de esta manera es igual a la que ya se muestra con una línea adicional donde se definió el número de ítems que deseamos mostrar usando el método `setVisibleItemCount`, si se definió un número de ítems menor al total de las opciones del ListBox se agregara un scroll que permite ver las demás opciones.

A continuación se muestra un ejemplo usando el ListBox que se creó anteriormente, donde el browser nos mostrará un aviso cada vez que se selecciona una opción del ListBox diciendo que opción seleccionamos. El código sería algo así:

```
11 public class FormWidgets implements EntryPoint {
12
13     public void onModuleLoad() {
14
15         final ListBox lb = new ListBox();
16         lb.addItem("Azul");
17         lb.addItem("Amarillo");
18         lb.addItem("Verde");
19         lb.addItem("Rojo");
20         lb.setSize("200px", "100px");
21
22         RootPanel.get().add(lb);
23
24         lb.addChangeHandler(new ChangeHandler() {
25
26             @Override
27             public void onChange(ChangeEvent event) {
28                 // TODO Auto-generated method stub
29                 int item = lb.getSelectedIndex();
30                 Window.alert(lb.getItemText(item));
31             }
32         });
33     }
34 }
```

Gráfico II.29: Creación del ListBox (segunda manera)

TextBox

El TextBox es la equivalencia en GWT a la etiqueta input de HTML, la cual es la forma estándar de capturar texto del usuario. Es un widget muy sencillo y fácil de usar. Se crea un TextBox de la siguiente manera:

```
TextBox nombre = new TextBox();
```

Gráfico II.30: Creación del widget TextBox

A continuación se muestra un ejemplo de una aplicación donde se pide el nombre al usuario y al presionar la tecla enter el browser muestra una ventana con el nombre que digito el usuario, el código sería este:

```
14 public class FormWidgets implements EntryPoint, KeyPressHandler {
15
16     public void onModuleLoad() {
17
18         Label lblnombre = new Label("Nombre");
19         TextBox txtnombre = new TextBox();
20         txtnombre.addKeyPressHandler(this);
21
22         RootPanel.get().add(lblnombre);
23         RootPanel.get().add(txtnombre);
24
25     }
26
27     @Override
28     public void onKeyPress(KeyPressEvent event) {
29         // TODO Auto-generated method stub
30
31         char a = event.getCharCode();
32         String nombre = ((TextBox)event.getSource()).getText();
33
34         if (a == KeyCodes.KEY_ENTER) {
35
36             Window.alert("Saludos "+nombre);
37
38         }
39     }
40 }
```

Gráfico II.31: Ejemplo de cómo utilizar un TextBox

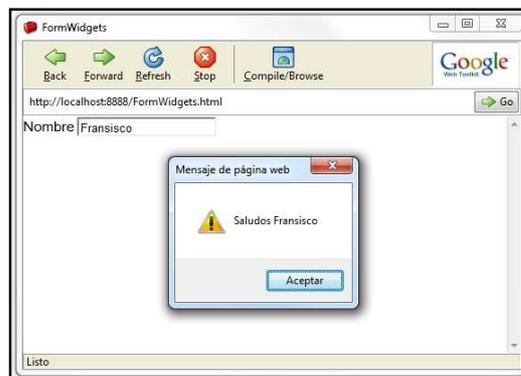


Gráfico II.32: Modo gráfico del TextBox

PasswordTextBox

Este widget funciona exactamente igual que el TextBox que se acabó de ver con la diferencia que su contenido está oculto del usuario para proteger datos como contraseñas.

Normalmente este widget se usa en un formulario de login. A continuación se muestra como se vería en una aplicación el PasswordTextBox.

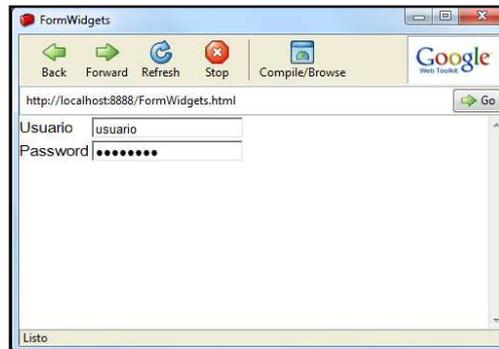


Gráfico II.33: Modo gráfico del widget PasswordTextBox

TextArea

El TextArea es otro widget que permite capturar texto del usuario, es parecido al TextBox con la diferencia que permite textos de múltiples líneas mientras que el TextBox nada más nos permite una sola línea. A continuación se verá un ejemplo del uso de un TextArea.

```
14 public class FormWidgets implements EntryPoint {
15
16     public void onModuleLoad() {
17
18         TextArea blog = new TextArea();
19
20         blog.setCharacterWidth(20);
21         blog.setVisibleLines(5);
22
23         RootPanel.get().add(blog);
24
25     }
26 }
```

Gráfico II.34: Creación del widget TextArea



Gráfico II.35: Modo gráfico del widget TextArea

RichTextArea

Los anteriores widgets de captura de texto nos permitían usar texto simple, es decir texto plano sin formato, pero hay caso en los que se quiere usar texto en negrilla o de algún color en especial u otros tipos de formato.

Para esto se usa el widget RichTextArea, que como su nombre lo indica es un TextArea pero que maneja texto enriquecido pudiendo decir, que se maneja de manera similar al TextArea, tiene los métodos `getText` y `setText` que permiten recuperar o asignar texto plano, pero además tiene 2 métodos adicionales que son `getHTML` y `setHTML` que nos permiten usar texto en formato HTML para así poder utilizar texto formateado. Este es un ejemplo de un RichTextArea:

I just put the **finishing touches** on the Blog Editor chapter. It looks great. A great illustration of how to interact with web services with GWT.

Gráfico II.36: Ejemplo del widget RichTextArea

FileUpload

El FileUpload widget solo funciona del lado del cliente por lo que permite al usuario seleccionar un archivo mas no enviarlo al servidor, ya eso depende en implementar la funcionalidad que da el widget.

La mayoría de navegadores tienen muchas restricciones de seguridad en cuanto al envío de archivos se refiere. Lo último implicaría un refresco de la página, pero la idea de GWT es que todo sea AJAX por lo que brinda un panel llamado FormPanel, que permite incluir el FileUpload y enviar archivos al servidor de manera asíncrona y cumpliendo con la metodología AJAX. Un FileUpload se vería más o menos así:

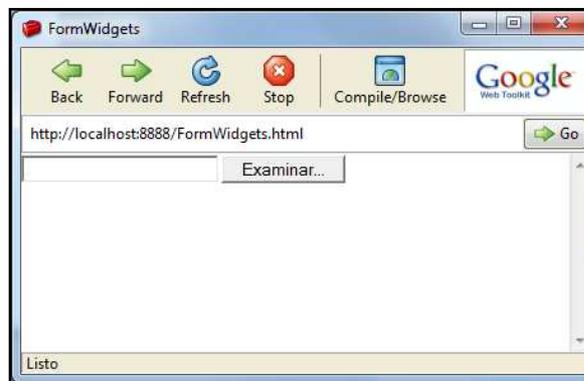


Gráfico II.37: Modo gráfico de un FileUpload

Widgets Complejos

Cuando los navegadores empezaron a aceptar JavaScript y HTML dinámico, se empezó a ver interfaces de usuario compuestas por objetos los cuales no tenían una etiqueta HTML equivalente.

Estos objetos fueron creados por la composición de etiquetas y el manejo de los eventos fue realizado con JavaScript. Estos widgets también se pueden crear con GWT, además que ya vienen algunos con el framework, estos son a los que llamamos widgets complejos. En esta categoría encontramos al Tree y al MenuBar

Tree

El Tree es parecido a los controles Tree de aplicaciones de escritorio, permite mostrar una vista jerárquica de datos que se puede contraer y colapsar. Los elementos que se muestra en un Tree pueden ser TreeItem, a los que se puede llamar ramas del árbol. Se puede agregar TreeItem a los TreeItem para agregar subniveles al árbol.

También se puede agregar una cadena u otros widgets al árbol. Instanciar un Tree widget y agregarlo a un panel es igual de sencillo que los demás widgets:

```
Tree arbol = new Tree();  
RootPanel.get().add(arbol);
```

Gráfico II.38: Creación de un TreeItem

A continuación se muestra un ejemplo agregando los ítems del Tree manualmente, pero se puede hacer esto dinámicamente con datos que se traiga de una base de datos que se encuentra en el servidor.

```
18 public void onModuleLoad() {  
19  
20     Tree arbol = new Tree();  
21     TreeItem rama1 = new TreeItem("Rama 1");  
22     rama1.addItem("subnivel rama 1");  
23  
24     TreeItem rama2 = new TreeItem("Rama 2");  
25     rama2.addItem(new CheckBox("Chekeo"));  
26  
27     TreeItem rama3 = new TreeItem("Rama 3");  
28     rama3.addItem(new Button("Clic"));  
29  
30     TreeItem rama4 = new TreeItem("Rama 4");  
31     rama4.addItem(new TreeItem("subnivel Rama 4"));  
32  
33     arbol.addItem(rama1);  
34     arbol.addItem(rama2);  
35     arbol.addItem(rama3);  
36     arbol.addItem(rama4);  
37  
38     RootPanel.get().add(arbol);  
39 }  
40 }
```

Gráfico II.39: Creación del widget TreeItem

En el ejemplo se ve que en la línea 20 se instancia el Tree, luego se definió 4 TreeItem cada uno de los cuales se le agregó un ítem ya sea una cadena, un widget u otro TreeItem. El resultado de ese ejemplo sería el siguiente:

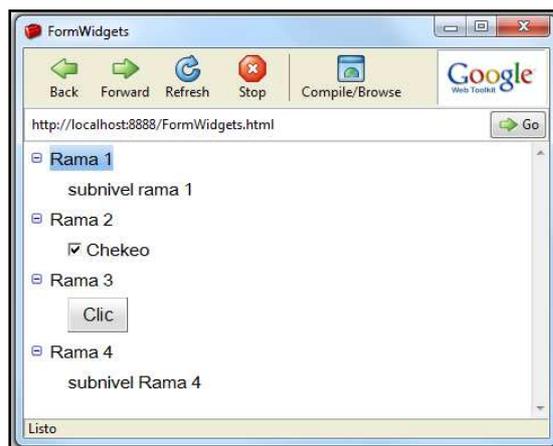


Gráfico II.40: Modo gráfico del widget TreeItem

MenuBar

Otro widget que entra en la categoría de widgets complejos es este llamado MenuBar. Como su nombre lo supone, es la equivalencia a los menús de las aplicaciones de escritorio. Tiene cierta similitud con el Tree que se vio, con la diferencia que solo se puede ver una subcategoría al tiempo. Normalmente se usa este widget para mostrar una lista de opciones dentro de la aplicación. Para instanciar un MenuBar solo falta una línea:

```
MenuBar menu = new MenuBar();
```

Gráfico II.41: Creación del widget MenuBar

También se puede pasar como parámetro otro MenuBar para llamar a un submenú. Además, se realiza llamadas a una clase MenuItem que funciona de manera similar al TreeItem que ya se ha visto. Ahora se ve un ejemplo de un menú sencillo.

```
18 public class FormWidgets implements EntryPoint {
19
20     public void onModuleLoad() {
21
22         Command cmd = new Command() {
23             public void execute() {
24                 //code to handle the command goes here
25             }
26         };
27     };
28
29     MenuBar menu = new MenuBar();
30
31     MenuBar ver = new MenuBar(true);
32     ver.addItem("Ver", cmd);
33     ver.addItem("Buscar", cmd);
34
35     MenuBar archivo = new MenuBar(true);
36     archivo.addItem("Nuevo", cmd);
37     archivo.addItem("Abrir", cmd);
38     archivo.addItem("Ver", ver);
39
40     MenuBar ayuda = new MenuBar(true);
41     ayuda.addItem("Acerca de ", cmd);
42
43     menu.addItem("Archivo", archivo);
44     menu.addItem("Ayuda", ayuda);
45
46     RootPanel.get().add(menu);
47 }
48 }
```

Gráfico II.42: Creación del widget MenuBar

En este ejemplo se define un objeto de tipo Command vacío, es decir con el método execute que es el que se ejecuta cuando se selecciona un ítem del menú, además se define el menú principal llamado "menu". Luego se define los submenús que se va a agregar al menú principal, se ve que los submenús se los instancia con el parámetro true para que se despliegue de manera vertical. Finalmente se agrega el menú principal al panel. El resultado sería la siguiente aplicación:



Gráfico II.43: Modo gráfico del MenuBar

2.5.2.14.2. Events y Listeners

Los eventos en Google Web Toolkit usan interfaces listener (de escucha de eventos), de una manera muy similar a como se utilizan en otros frameworks. Una interfaz *listener*, define uno o más métodos que los widgets usan para anunciar un evento. Una clase que recibe eventos de un tipo en particular, implementa la interfaz listener asociada con el tipo de eventos que recibe y luego pasa una referencia del widget que generó el evento para "suscribirlo" a un conjunto de eventos.

La clase Button, por ejemplo, publica eventos click. La interfaz asociada para este tipo de eventos es ClickListener:

```
public void ejemploDeClickListener() {  
    Button b = new Button("Dame clic");  
    b.addClickListener(new ClickListener() {  
        public void onClick(Widget sender) {  
            // aquí se determina que sucede cuando se hace clic  
        } });  
}
```

Los widgets proveen su apuntador this como el parámetro sender cuando invocan un método listener, permitiendo que sea sencillo para el listener distinguir entre los diferentes publicadores. Esto hace un mejor uso de la memoria, pero requiere más líneas de código para su implementación, como en el siguiente ejemplo:

```
public class EjemploListener extends Composite implements ClickListener {  
    private FlowPanel fp = new FlowPanel();  
    private Button b1 = new Button("Button 1");  
    private Button b2 = new Button("Button 2");  
    public EjemploListener() {  
        initWidget(fp);  
        fp.add(b1);  
        fp.add(b2);  
        b1.addClickListener(this);  
        b2.addClickListener(this);  
    }  
    public void onClick(Widget sender) {  
        if (sender == b1) {  
            // handle b1 being clicked  
        } else if (sender == b2) {
```

```
// handle b2 being clicked  
}}}
```

2.5.2.14.3. Creando widget personalizados

Se ha visto el uso de varios de los widgets que provee GWT en su framework. Ahora se verá lo que son los widgets compuestos o “composite widgets”. Los “*composite widgets*” son elementos potentes de GWT los cuales son widgets creados por nosotros combinando los widgets y paneles que ya se ha visto anteriormente. Esto es muy útil si se quiere crear controles de usuario personalizados y reutilizables.

Se puede considerar a estos widgets compuestos como mini aplicaciones GWT, ya que están compuestas por widgets y proporcionan una funcionalidad dada por el creador del control.

Para crear un widget compuesto lo que se hace es crear una clase y hacer que herede de “Composite”, la implementación de interfaces depende de los eventos que se vaya a manejar en el widget. Luego como atributos de la clase se agrega los widgets que se vaya a utilizar.

A continuación se ve un ejemplo de un widget que estará compuesto de un TextBox y un Button, y que al dar clic en el botón se muestra lo escrito en el TextBox en un mensaje del browser.

```
8 public class WidgetCompuesto extends Composite implements ClickHandler {
9
10     private TextBox texto = new TextBox();
11     private Button btnver = new Button("Ver");
12
13     public WidgetCompuesto() {
14
15         texto.setHeight("30px");
16
17         btnver.addClickHandler(this);
18         btnver.setHeight("30px");
19
20         FlowPanel panel = new FlowPanel();
21         panel.add(texto);
22         panel.add(btnver);
23
24         this.initWidget(panel);
25     }
26
27     @Override
28     public void onClick(ClickEvent event) {
29         // TODO Auto-generated method stub
30         Window.alert(texto.getText());
31     }
32
33 }
```

Gráfico II.44: Ejemplo de widget compuesto

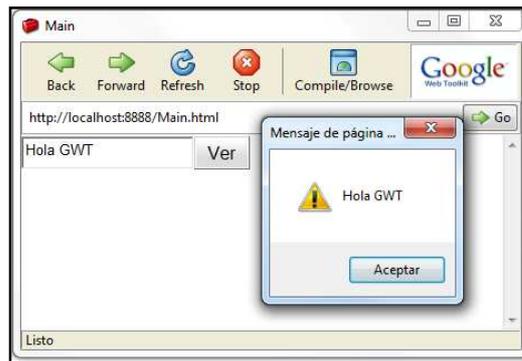


Gráfico II.45: Modo gráfico del widget compuesto

Pasos para armar los Widgets compuestos

1. **Identificar los widgets:** como se sabe los widgets compuestos son hechos de varios widgets los cuales pueden ser los estándares de GWT u otro widget compuesto, por eso es bueno que antes de construir el widget se identifique los widgets que se va a incluir en él.
2. **Elegir el diseño:** después de elegir los widgets que se usará se debe definir el diseño a usar, es decir la disposición de los widgets dentro de nuestro control.

3. **Identificar interfaces:** esto se refiere a las interfaces que se va a implementar en el widget para el manejo de los eventos, depende de las necesidades y de las funcionalidades que se vaya a implementar. Se implementará las interfaces que sean útiles para el propósito.
4. **Construcción del widget:** es la construcción del widget como tal, ya que se ha definido la estructura es hora de implementar dicha estructura. Este paso tiene sub-etapas que son:
 - a. Implementar las interfaces identificadas.
 - b. Implementar métodos adicionales a los requeridos por las interfaces que implementamos.
 - c. Crear la estructura del widget y establecer el manejo de eventos.
 - d. Implementar el código que maneja los eventos esperados.
5. **Establecer los estilos:** en esta etapa más que todo lo se hace es establecer una convención para los nombres de estilo de cada componente widget o panel.
6. **Pruebas:** como todo desarrollo de software la etapa final es la prueba del widget. la idea es que el widget se comporte bien y haga lo que tiene que hacer para evitar sorpresas.

2.5.2.14.4. Hojas de estilos (CSS)

Los widgets de GWT confían su estilo visual a hojas de estilo en cascada. Cada widget tiene un nombre de estilo asociado que lo enlaza a una regla CSS. Un nombre de estilo de widget es asignado usando el método `setStyleName()`. Por ejemplo, `Button` tiene como nombre de estilo por defecto `gwt-Button`. En ese orden de ideas, si se deseara por ejemplo

asignar un tamaño específico a la *letra* (fuente) de todos los botones de la aplicación, se podría usar la siguiente regla en tu archivo CSS:

```
.gwt-Button { font-size: 150%; }
```

Estilos complejos

Algunos widgets tienen asociados tipos de estilos más complejos. MenuBar, por ejemplo, tiene los siguientes estilos:

```
.gwt-MenuBar { the menu bar itself }  
.gwt-MenuBar .gwt-MenuItem { menu items }  
.gwt-MenuBar .gwt-MenuItem-selected { selected menu items }
```

En este ejemplo, se tienen dos reglas que aplican a los ítems de los menús. El primero aplica a todos los ítems de los menús, mientras el segundo (con el sufijo *-selected*) aplica solo para el ítem de menú que esté seleccionado. Un nombre de estilo de ítem de menú seleccionado será asignado a “*gwt-MenuItem gwt-MenuItem-selected*”, especificando que ambas reglas de estilo serán aplicadas. La manera más común de hacer esto es usar `setStyleName` para asignar el estilo de nombre base, luego `addStyleName()` y `removeStyleName()` para añadir y remover el segundo nombre de estilo.

Archivos CSS

Generalmente, las hojas de estilo son situadas en paquetes que son parte del módulo *public path*. Entonces simplemente incluya una referencia a la hoja de estilo en la página HTML, así:

```
<link rel="stylesheet" href="mystyles.css" type="text/css">
```

2.5.2.14.5. Atado de imágenes (image bundle)

Generalmente, una aplicación utiliza muchas imágenes pequeñas para iconos. Una petición al servidor debe ser enviada por cada una de las imágenes, y en algunos casos, el tamaño de la imagen es menor que la cabecera de la petición en sí, que es enviada con los datos de la imagen.

Una imagen atada, es un conjunto de imágenes dentro de una sola imagen, con una interfaz para acceder a las imágenes individualmente. De esta forma, en vez de hacer una petición al servidor por cada imagen que se necesita, la aplicación realiza solo una petición y recupera todas las imágenes.

Ya que los nombres de archivo de la composición de imágenes están basados en un *hash* del contenido del archivo, el nombre de archivo cambiará solamente si la imagen es cambiada. Esto significa que es seguro para los clientes guardar en caché la composición de imágenes permanentemente, lo que evita estar verificando innecesariamente si las imágenes han sufrido cambios en el servidor. Para lograr esto, la configuración del servidor necesita especificar que la composición de imágenes nunca expira. Además de la velocidad de carga de la aplicación, usar image bundle previene el efecto “bouncy” al cargar las imágenes en el navegador.

Especificaciones

- Creando y usando image bundle.- Define un *image bundle*
- Image bundle y localización.- Crea *locale-sensitive image bundle* para usar las capacidades de localización de GWT.

Crear y Usar Image Bundle

Para definir un *image bundle*, el usuario necesita extender la interfaz `ImageBundle`. La interfaz derivada puede tener cero o más métodos, donde cada método:

- No toma parámetros
- Tiene un tipo de retorno `AbstractImagePrototype`, y
- Puede tener un metadata `gwt.resource` opcional lo que especifica el nombre del archivo de imagen en el classpath del módulo.

Los tipos de archivo válidos son png, gif y jpg. En caso de que haya múltiples archivos de imagen con diferentes extensiones, el orden de precedencia de extensiones es (1) png, (2) gif y (3) jpg.

Un `ImageBundle` para los iconos de un procesador de texto podría ser definido así:

```
public interface WordProcessorImageBundle extends ImageBundle {  
    public AbstractImagePrototype new_file_icon();  
    public AbstractImagePrototype openFileIcon();  
    public AbstractImagePrototype saveFileIcon();  
}
```

Los métodos en un *image bundle* retornan objetos *AbstractImagePrototype* (en vez de objetos *Image*, como se podría esperar), ya que los objetos *AbstractImagePrototype* proveen un aligeramiento adicional del peso de la imagen.

El siguiente ejemplo muestra cómo usar el *image bundle* que se definió en la aplicación:

```
public void useImageBundle() {
    WordProcessorImageBundle wpImageBundle = (WordProcessorImageBundle)
    GWT.create(WordProcessorImageBundle.class);
    HorizontalPanel tbPanel = new HorizontalPanel();
    tbPanel.add(wpImageBundle.new_file_icon().createImage());
    tbPanel.add(wpImageBundle.openFileIcon().createImage());
    tbPanel.add(wpImageBundle.saveFileIcon().createImage());
}
```

2.5.2.15. RPC (Remote Procedure Call)

Hasta la aparición de AJAX, cada vez que una aplicación web tenía que realizar alguna operación en el servidor, se tenía que enviar una petición (GET o POST) a este, que tras efectuar la operación devolvía un documento HTML completo, que sustituía al anterior.

Con AJAX podemos hacer peticiones al servidor sin sustituir el documento actual y además, podemos optimizar la respuesta haciendo que el servidor devuelva únicamente datos (sin incluir formato). Todo esto además de forma asíncrona por lo que se mejora mucho la experiencia de usuario. GWT dispone de un mecanismo por el cual realizar este tipo de peticiones resulta muy sencillo. Este mecanismo ya existía en las antiguas arquitecturas cliente/servidor y se llama RPC (Remote Procedure Call).

2.5.2.15.1. Definición de RPC

Esto se trata de la comunicación entre el Cliente y el Servidor. La tecnología de GWT permite implementar del lado del Servidor distintas tecnologías como PHP, o ASP.NET. Esto es RPC (Remote Procedure Call) que viene siendo en español “llamada a procedimiento remoto” o Servlets Java, que pueden estar residiendo en un Tomcat u otro Servidor de *Servlets* del Mercado. El RPC en Google Web Toolkit permite fácilmente al cliente enviar y recibir objetos de Java sobre HTTP.

El concepto de RPC es muy similar al RMI. Una diferencia fundamental entre Google Web Toolkit y las tradicionales aplicaciones web, es que las aplicaciones Google Web Toolkit no necesitan de otras páginas web mientras son ejecutadas. Ya que las páginas construidas con Google Web Toolkit corren como aplicaciones sobre el navegador, éstas no necesitan hacer nuevas peticiones al servidor, por ejemplo, para realizar actualizaciones en la interfaz de usuario.

Cuando es usado adecuadamente, RPC da la oportunidad de mover toda la lógica de la interfaz de usuario al cliente, lo que mejora el funcionamiento de la aplicación, reduce el ancho de banda usado, la carga al servidor, y presenta al usuario final una experiencia más agradable navegando por la página ¹⁰.

En la siguiente imagen se muestra un bosquejo de una comunicación entre un servidor y el cliente.

¹⁰ Por: Página oficial de GWT

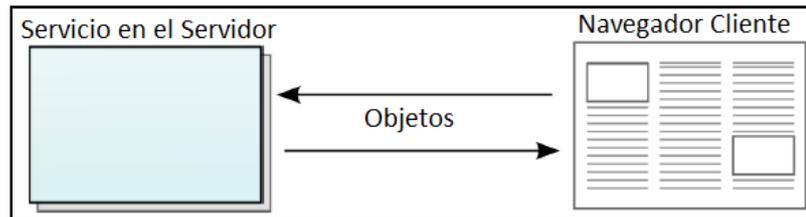


Gráfico II.46: Comunicación entre el cliente y el servidor RPC

2.5.2.15.2. Ventajas del RPC-GWT

Entre las principales ventajas que tenemos de los RPC-GWT son:

- Primero permite mantener todo el proyecto en un único lenguaje JAVA proporcionando cierta claridad y coherencia al código, además de poder programar todo en el mismo IDE y en el mismo lenguaje.
- Segundo maneja automáticamente el transporte de los Objetos Java entre el Cliente y el Servidor, realizando el proceso que se conoce como Serialización/Deserialización en forma transparente para el programador, a diferencia de XMLHttpRequest con el que podemos intercambiar solamente datos de texto (XML, JSON o lo que sea) que se pueden o no convertir a un objeto en el cliente, y eso agrega otro paso más que es innecesario con RPC.
- Tercero y muy importante, con RPC se puede depurar el código que se encuentra del lado del servidor desde el mismo IDE.

2.5.2.15.3. Desventaja del RPC-GWT

- RPC requiere que el código del servidor se encuentre corriendo dentro de un contenedor J2EE (son servlets) y no siempre tenemos uno a la mano. Normalmente los proveedores

de hosting incluyen en sus planes php pero no un contenedor J2EE a menos que este servicio sea específicamente contratado, lo que puede complicarse al momento de querer instalar la aplicación en un servidor fuera de nuestra computadora.

2.5.2.15.4. Anatomía de RPC en GWT

Cada servicio tiene una pequeña familia de interfaces y clases. Algunas de esas clases, como el servicio Proxy, son generados automáticamente y generalmente nunca se percata de que existen. El patrón de clases es idéntico para todos los servicios que implementa, por lo que es una buena idea familiarizarse un poco con los términos y el propósito de cada capa, en el procesamiento de llamadas al servidor.

El código del lado del servidor que es invocado desde el cliente es frecuentemente llamado “servicio”, por lo que el “llamar procedimientos remotos” es comúnmente llamado como invocación de servicios. Aunque es necesario tener claro que el término “servicio” en este contexto NO tiene el mismo concepto de “web-service”. En realidad, los servicios Google Web Toolkit no son lo mismo que “Simple Object Access Protocol” (SOAP).

Se puede decir que hay 3 partes o piezas que se debe ensamblar para poner a funcionar una aplicación que use RPC. **La primera:** es el servicio que corre en el servidor, **la segunda:** es el cliente en el browser que llama al servicio, y **la tercera:** son los objetos que son transportados entre el cliente y el servidor. Tanto el cliente como el servidor tienen la habilidad de serializar y deserializar datos, por lo que los objetos pueden ser enviados entre ambos como texto ordinario.

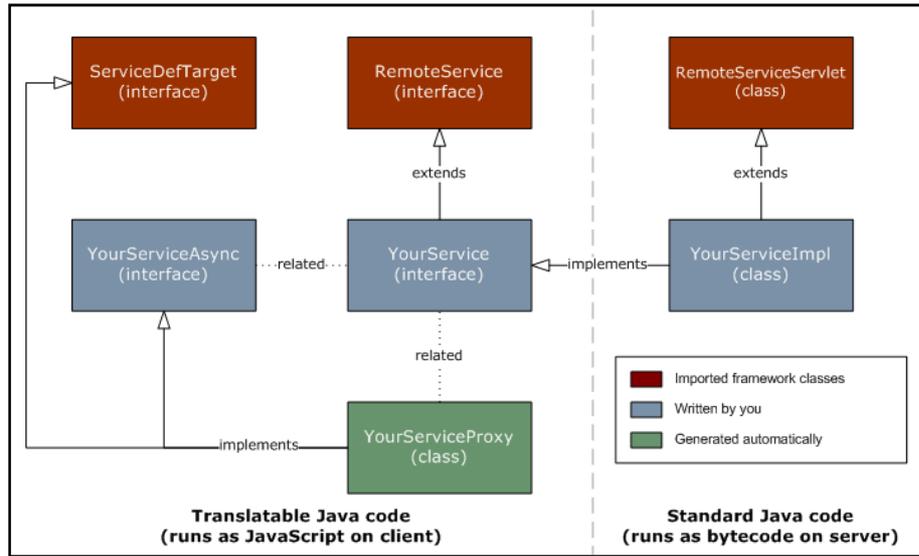


Gráfico II.47: Anatomía de RPC en GWT

A continuación se va a explicar estas partes, comenzando por los objetos y un poco de la serialización.

Primera parte: Los Objetos

Se empieza con los objetos porque los datos son los que le dan vida al GWT-RPC. Como ya se dijo RPC consiste en llamar métodos que se encuentran en un servicio en el servidor el cual enviará el resultado de la ejecución del método hacia el cliente.

Como cualquier método de Java se puede pasar parámetros a los métodos los cuales pueden ser tipos primitivos u objetos. Ya se dijo anteriormente que los objetos se pasan serializados entre el cliente y el servicio así que hay límites entre los tipos de datos que se pueden enviar, los cuales están en la siguiente tabla:

Tabla II.2: Tipos de datos que usa RPC

Tipos primitivos	boolean, byte, char, double, float , int, long , shot
Contenedores de los tipos primitivos	Boolean, Byte, Character, Double, Integer, Float...
Subconjuntos de objetos java	Solo ArrayList, Date, HashMap, HashSet, String, Vector
Clases definidas por el usuario	Cualquier clase que implemente IsSerializable
Arreglos	Arreglos de cualquier tipo serializable

Los tipos de datos propios de java son limitados y son los incluidos en la librería de GWT que emula la JRE de java.

Con respecto a los otros tipos de datos definidos por el usuario deben implementar la interface *IsSerializable*, la cual no tiene métodos que implementar y que se utiliza para que GWT sepa que ese objeto podrá ser serializado. Las clases implementarán la interface *com.google.gwt.user.client.rpc.IsSerializable*, todos los atributos que no tengan el modificador *transient* son serializables, y además la clase tendrá un constructor sin parámetros o no tendrá constructor.

Segundo paso: El Servicio

El próximo paso para usar RPC es definir e implementar el servicio que será ejecutado en el servidor. Esto consistirá en una interface de java, la cual describiría el servicio y la implementación del mismo.

Para definir el servicio se necesita crear una interface y hacer que herede de la interface que provee GWT llamada *RemoteService*. Sería algo como esto:

```
9 public interface Servicio extends RemoteService {  
10  
11     public String getFecha();  
12  
13 }
```

Gráfico II.48: Crear un servicio RPC

Como se ve es tan sencillo como definir el nombre de la interface, heredar de la interface *com.google.gwt.user.client.rpc.RemoteService*, definir los métodos que se desea en el servicio teniendo en cuenta que los parámetros y los valores de retorno deben ser serializables.

Esta interface se la debe colocar en el paquete “cliente” de nuestro modulo. Aunque si se usa algún IDE de desarrollo como eclipse con algún plugin este se encargará de todo esto, ya se verá después como hacer esto con eclipse.

Ya se ha creado la interface para el servicio, ahora es momento de implementar sus métodos. Se hace esto creando un servlet que herede de *RemoteServiceServlet* y que implemente nuestra interface de servicio. El código sería algo como esto:

```
6 public class ServicioImpl extends RemoteServiceServlet implements Servicio {
7
8     @Override
9     public String getFecha() {
10         // TODO Auto-generated method stub
11         return "01/01/2010";
12     }
13
14 }
```

Gráfico II.49: Creación del servlet para el servicioRPC

No hay mucho que explicar, en el servlet se implementa los métodos del servicio que se desea crear. Se ve que el nombre del servlet es el mismo que el de la interface con el sufijo Impl el cual es notación de GWT para reconocer a ese servlet como la implementación del servicio que define la interface que se creó anteriormente.

Cuando se llama al servicio desde el cliente GWT hace la mayor parte del trabajo por nosotros, sin embargo todavía se necesita crear una última interface. Esta interface será usada por el compilador de GWT cuando este genere el objeto proxy del servicio.

Tercer paso: El cliente

Un objeto proxy es la instancia de un objeto que reenvía la petición a otro objetivo. En este caso se estará llamando a un método local y el objeto proxy será el encargado de serializar los parámetros, llamar al servicio remoto, y manejar la deserialización del resultado. Todo esto lo hace GWT por nosotros. En el código del lado del cliente se crea el objeto proxy con la siguiente línea:

```
GWT.create(Servicio.class);
```

Gráfico II.50: Creación del Objeto Proxy

Aquí se llama al método estático *create* de la clase *com.google.gwt.core.client.GWT* pasándole como parámetro la clase de la interface del servicio. Esto devuelve un objeto proxy el cual se puede usar para asignar la URL del servicio y llamar los métodos del servicio remoto. El objeto proxy implementará 2 interfaces, una que se debe crear por nosotros y otra que provee GWT.

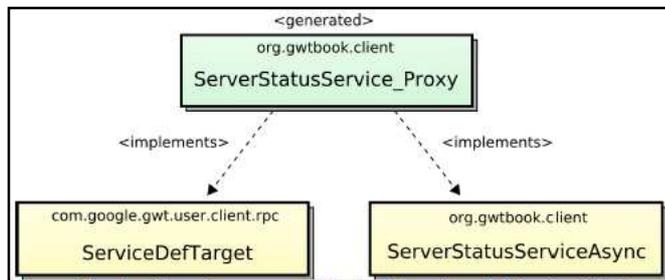


Gráfico II.51: Ejemplo del Objeto proxy

En la figura anterior se ve un ejemplo con un servicio llamado *ServerStatus*, la interface *ServiceDefTarget* es la que viene con GWT, incluye el método *setServiceEntryPoint* para especificar la URL del servicio. La segunda es la que se define por nosotros y se nos provee métodos asíncronos para llamar el servicio remoto. Como se ve es el nombre del servicio con el sufijo *Async*.

Los métodos en esta interface deben coincidir con los nombres de los métodos de la interface de servicio pero cambiando algo en la firma de los métodos, el valor de retorno debe ser *void* y además se debe agregar un parámetro de tipo *com.google.gwt.user.client.rpc.AsyncCallback*. En la siguiente tabla se muestra un ejemplo de esto:

Tabla II.3: Ejemplo del servicio Async

Service interface	Asynchronous service interface
String methodOne (int i);	void methodOne (int i, AsyncCallback cb);
List methodTwo();	void methodTwo (AsyncCallback cb);
Boolean methodThree (int a, int x);	void methodThree (int i, int x, AsyncCallback cb);

Esta interface es usada del lado del cliente por lo que debe ser ubicada en el paquete “cliente”.

2.5.2.15.5. Crear servicios RPC

Para poder realizar el servicio RPC se va a seguir los siguientes pasos:

- a. Sobre el paquete *client* de nuestro proyecto se hace click con el boton derecho del mouse y presionamos New – Other.
- b. Se elige la carpeta de Cypal Studio y dentro de esta se elige *GWT Remote Service*. Y damos click en el botón **Next**.

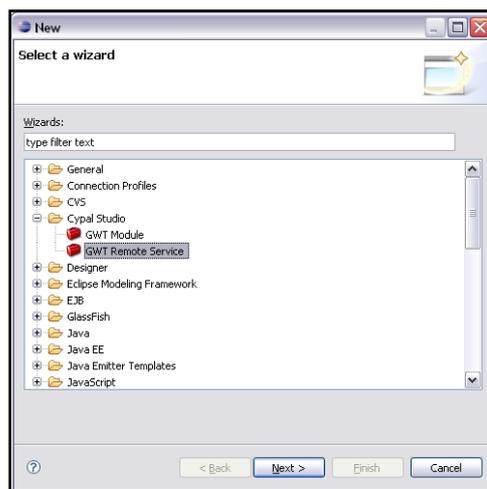


Gráfico II.52: Primera pantalla para crear un servicio RPC

- c. En la siguiente pantalla del wizard se le da el nombre al servicio, en este caso será REE_Service que es el nombre de nuestra interfaz. Además se debe indicar el nombre del URI que le va a corresponder al servicio. En nuestro caso se le escribe /REE_Service. Luego se presiona **Finish**, y queda así el servicio RPC creado.

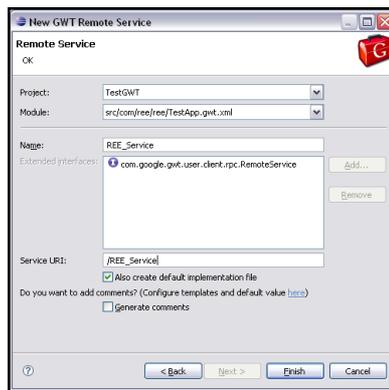


Gráfico II.53: Segunda pantalla para crear un servicio RPC

- d. Se comprueba que el servicio RPC se ha creado correctamente, por lo que se tendría que tener 2 nuevas interfaces en el paquete *client*, con los nombres que ya se indicó, y además una clase con el nombre **REE_ServiceImpl** que cuelga del paquete server.

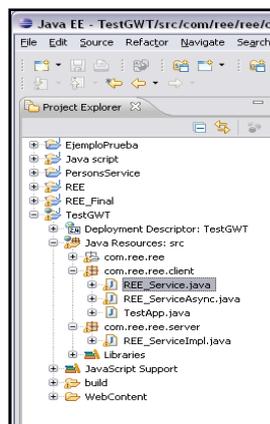


Gráfico II.54: Visualización de las interfaces del servicio RPC

Ninguna de estas interfaces / clases va a tener métodos al momento salvo unos métodos de utilidad que agrega Cypal en la interfaz principal para obtener la instancia de la interfaz remota.

Ahora se agrega un método al servicio, para ello abrimos la interfaz generada REE_Service y agregamos lo siguiente:

```
public String info(String text);
```

Se graba la interfaz y se ve que ocurren 2 cosas. *Primero* la clase que implementa en el *server* nos indica con una cruz roja que tiene un error, esto es porque hemos creado un método en *REE_Service* y aun no lo hemos implementado.

Esto se lo resuelve fácilmente, se abre la clase java y haciendo click con el botón derecho encima de la clase *REE_ServiceImpl* se selecciona la opción *Source* y dentro de ella la opción *Override/Implement Method*. Ahí se ve que se nos sugiere sobrescribir el método *info* de la interfaz *REE_Service*, que es justamente lo que se quiere. Damos **ok** y obtenemos depurados los errores del método deseado.

Luego se modifica el método por:

```
public String info(String data) {  
    return "Usted envió el texto:" + data;  
}
```

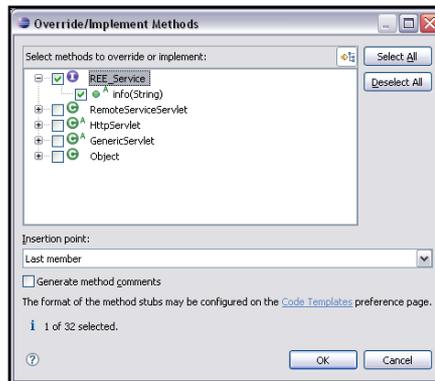


Gráfico II.55: Sobrescribir la clase REE_ServiceImpl

e. Por último se graba la modificación y ya tenemos el servicio listo para ser invocado desde el cliente.

2.5.2.15.6. Invocar servicios

Ahora que se ha definido todas las interfaces necesarias se puede realizar el llamado al servicio desde el cliente y para esto se debe hacer lo siguiente:

- Instanciar un objeto proxy como ya vimos.
- Especificar la URL del servicio.
- Crear un método *callback* para manejar el resultado de la llamada asíncrona del método.
- Llamar al método.

Instanciar el objeto proxy

Se hace esto llamando al método estático `GWT.create()` pasándole como parámetro la clase de la interface del servicio que se creó. Luego se necesita realizar un casting de esto objeto a la interface asíncrona que ya se ha creado anteriormente.

```
ServerStatusServiceAsync serviceProxy =  
(ServerStatusServiceAsync) GWT.create(ServerStatusService.class);
```

Gráfico II.56: Instanciar el objeto proxy

Especificar la URL del servicio

Para este paso se necesita realizar otro casting del objeto proxy que se captura al tipo *ServiceDefTarget* para de esta manera definir la URL del servicio remoto.

```
ServiceDefTarget target = (ServiceDefTarget) serviceProxy;  
target.setServiceEntryPoint(GWT.getModuleBaseURL() + "server-status");
```

Gráfico II.57: Especificar la URL del servicio

La URL debe ser la misma que está en la definición del *servlet* en el *web.xml*.

Crear un objeto callback

En este paso se crea un objeto *callback*, este objeto implementará la interface *com.google.gwt.user.client.rpc.AsyncCallback* y será ejecutado cuando el resultado sea recibido del servidor, como ya se ha añadido un parámetro *AsyncCallback* en los parámetros de los métodos de la interface asíncrona que se definió, este objeto será el parámetro extra que se envía en el llamado a la función.

Para demostrar esto con un ejemplo se va a crear un objeto anónimo que implemente la interface *AsyncCallback* como se muestra en el siguiente gráfico.

```
AsyncCallback callback = new AsyncCallback() {  
  
    public void onFailure (Throwable caught) {  
        GWT.log("RPC error", caught);  
    }  
  
    public void onSuccess (Object result) {  
        GWT.log("RPC success", null);  
    }  
  
};
```

Gráfico II.58: Crear un Objeto CallBack

La interface *AsyncCallback* tiene 2 métodos que deben ser implementados que son: *onFailure* que se ejecuta cuando hay algún error en la ejecución del método. Y el método *onSuccess* cuando el método fue ejecutado exitosamente y el resultado fue recibido del servidor, en este caso el resultado es recibido por el método en el parámetro *result* el cual se puede usar para cualquier operación que se desee realizar.

Llamar al servicio remoto

El último paso es llamar al servicio, lo cual es bastante simple y consiste en una sola línea que sería:

```
serviceProxy.getStatusData(callback);
```

Gráfico II.59: Llamar al servicio remoto

2.5.2.16. Desplegar en Tomcat

Hay que recordar que para ejecutar una aplicación GWT tenemos dos opciones: "hosted mode" y "web mode". Para probar un servicio en "hosted mode", podemos hacer que la

propia shell de pruebas pueda ejecutar el Servlet. Para ello simplemente se tiene que declarar el Servlet en el fichero descriptor del módulo (Principal.gwt.xml) como se indica a continuación.

```
<servlet path="/REE_RPC" class="com.ree.ree.server.REE_RPCImpl"/>
```

De esta forma tan sencilla, se puede probar nuestra aplicación antes de desplegarla en un servidor. Para probar la aplicación en modo web, se debe crear un fichero WAR. Este fichero debe incluir:

- Clases compiladas del paquete "com.ree.ree.server" así como todas sus dependencias. Para evitar problemas en este sentido, se incluye todas los .class de la aplicación, aunque se sepa de antemano que clases no se van a ejecutar en la JVM del servidor.
- Los recursos estáticos que usa la aplicación (imágenes, css, js, etc) así como el resultado de la compilación GWT de las clases cliente. En definitiva, todo el contenido de la carpeta "www".
- Librería "gwt-user.jar" dentro de la carpeta "WEB-INF/Lib". Hay que tener en cuenta que si se despliega la aplicación en Tomcat, hay que tener cuidado al incluir la librería "gwt-user.jar" ya que esta contiene las clases de los paquetes "javax.servlet.*", cosa que no está permitida por Tomcat.
- Descriptor de despliegue "web.xml" donde se da de alta el Servlet "/REE_RPCImpl".

CAPÍTULO III ANÁLISIS COMPARATIVO

3.1. Introducción

Los IDEs (Integrated Development Environment) son un conjunto de herramientas para el programador, que suelen incluir en una misma suite, un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores y debuggers e integración con sistemas controladores de versiones o repositorios, permitiendo mucha más versatilidad para depurar los programas.

En el presente capítulo se proporciona un análisis comparativo sobre los IDE's más importantes que usa GWT para el desarrollo de sus aplicaciones web, sabiendo que GWT no tiene un IDE específico, puede utilizar cualquier entorno de desarrollo integrado que use el lenguaje de programación JAVA y para hacer efectivo su uso lo único que se debe hacer es referenciar en cada uno de los IDE's la dirección en donde se ha descomprimido GWT.

Teniendo como referencia lo expresado anteriormente, se procede a la determinación de las herramientas que se van a comparar en base a consultas sobre estudios realizados en el Internet sobre los IDE's más populares, potentes y fáciles de usar.

Una vez realizada esta selección se explica en detalle cada uno de los IDE's seleccionados por los autores para luego determinar y detallar los parámetros e indicadores que serán el punto de partida para ejecutar el análisis comparativo y determinar el entorno de desarrollo integrado más idóneo para la puesta en práctica de la parte aplicativa del presente proyecto.

Finalmente, se obtendrán los resultados finales a partir de la tabulación de la calificación asignada a cada uno de los indicadores de los parámetros de evaluación de los IDE's, se hará uso también de gráficos estadísticos que determinarán a simple vista el IDE más adecuado para el desarrollo del presente trabajo.

3.2. Determinación de las herramientas a comparar

En la actualidad existen muchos IDE's para desarrollar proyectos o aplicaciones web usando el lenguaje de programación JAVA, para este caso en particular usaremos el framework GWT como referencia para realizar la consulta respectiva en el Internet. A continuación se mostrará una serie de páginas web que se ha tomado como referencia para la elección del IDE más adecuado para el desarrollo del presente trabajo, en las que se expresa la popularidad y facilidad de uso que tiene para el programador.

- En el siguiente sitio web: <http://elcodigok.blogspot.com/2008/03/ides-para-programar-en-java.html> llamado “El CoDiGo K” publicado el 15 de marzo del 2008 por Daniel M. Maldonado analista en sistemas de la Universidad Católica de Santiago del Estero, habla de los IDE’s más populares para programar en Java, entre los cuales se encontró los que a continuación se detalla:
 - BlueJ
 - Eclipse
 - IntelliJ IDEA
 - Jbuilder
 - JCreator
 - JDeveloper
 - NetBeans
 - Sun Java Studio Enterprise

- En el sitio web llamado “INFORMATICA PURA” escrito por Snyflex el 14 Abril, 2010 y cuya URL es: <http://informaticapura.com/2010/04/14/el-ide-de-tu-preferencia/> se publicó sobre el IDE más adecuado para el desarrollo de aplicaciones web en Java en el que luego de un pequeño debate.

Y en base a cada una de las ventajas que estos pudieran tener como su costo se determinó que para la programación de Java los IDE más adecuados son:

- Eclipse

- NetBeans

- En un artículo publicado en la web, Teknoda sobre la “Selección de un IDE gratuito” el cual lo se puede encontrar tipiendo la URL:

<http://www.teknoda.com/tips/java/java01.pdf>, se lista una serie de IDE populares según dicho autor entre los cuales se encuentra los siguientes:

- Sun ONE Studio 4
- JEdit
- JCreator
- Eclipse
- NetBeans
- Sun ONE Studio 4 Enterprise Edition
- IntelliJ IDEA
- Metrowerks Code Warrior
- Borland JBuilder
- IBM Visual AGE for Java
- IBM WebSphere Development Studio Client

Donde las cinco primeras son open source (código abierto), mientras que las siguientes son licenciadas o comerciales. Al final dicho autor recomienda usar JEdit, Eclipse y NetBeans, por no tener costo y ser más conocidas a nivel de los programadores de Java.

- En un artículo publicado el 26 de Agosto del 2009 por Erik Burke sobre la “selección del IDE más adecuado para GWT”, el mismo que lo se puede encontrar en la dirección <http://stuffthathappens.com/blog/2009/08/26/gwt-ide-selection/>, nos recomienda usar:
 - Eclipse
 - NetBeans
 - IntelliJ IDEA

Los mismos que los ha evaluado en base a una serie de parámetros como: facilidad de instalación, facilidad de uso, soporte técnico, entre otros.

- En el sitio web publicado el 04 de Mayo del 2010, con el tema “El mejor IDE para Java” <http://comomedices.blogspot.com/2010/05/el-mejor-ide-para-java.html>. Se muestra los IDEs más populares y utilizados en el entorno de desarrollo de Java, en las que se encuentran los siguientes:
 - NetBeans
 - Eclipse
 - JBuilder
 - JDeveloper
 - IntelliJ IDEA

- En el siguiente PDF escrito por Cristian Castiblanco, en el año 2007, titulada “Tutorial (1.0) de Introducción a Google Web Toolkit” descargada del siguiente enlace:

http://www.google.com.ec/url?sa=t&source=web&cd=1&ved=0CBQQFjAA&url=http%3A%2F%2Fcasidiablohost.googlepages.com%2FIntroduccion-al-Google-Web-Toolkit.pdf&rct=j&q=Tutorial+%281.0%29+de+Introducci%C3%B3n+a+Google+Web+Toolkit&ei=Ez0pTOXVH4OCIAeG-IzhAw&usq=AFQjCNHW6Npt_gaA53GgLYmiGjo-GzS0mg, donde se indica los IDEs favoritos

y más usados que son los siguientes:

- Eclipse
 - IntelliJ IDEA
 - JProfiler
 - JUnit
-
- En el sitio web “builder au by developers for developers”
http://www.builderau.com.au/program/java/soa/Java_jams_five_IDEs_tested/0,339024620,339174040,00.htm, realizó un estudio comparativo de los principales IDEs:
 - Netbeans 4.0
 - Eclipse 3.0.1
 - WebSphere Studio Application Developer 5.1.2
 - Sun Studio Creator 4.26
 - Borland JBuilder 2005

Al final de todo el estudio comparativo, obtuvo los siguientes resultados sobre los IDEs más populares como se indica a continuación:

Tabla III.1: Resultados IDE Adecuado

IDE	Evaluación Total
Netbeans 4.0	25.5
Eclipse 3.01	26.5
IBM WebSphere Studio Application Developer 5.1.2	24.5
Sun Studio Creator 4.26	24.5
Borland JBuilder 2005	25.5

Una vez expresado las consultas anteriores sobre los IDE's más populares y adecuados que existen el mercado para el desarrollo de aplicaciones web usando el lenguaje de programación Java según los sitios web expresados, se va a seleccionar los siguientes entornos de desarrollo integrados: Eclipse, Netbeans e IntelliJ IDEA por las siguientes razones que a continuación se expresan:

- Porque tanto Eclipse, Netbeans e IntelliJ IDEA son los que más se repiten, es decir, gozan de mayor popularidad en casi todos los sitios web visitados anteriormente según sus diferentes autores que son desarrolladores de Java, además se cuenta con la información necesaria para su respectiva utilización.
- Los IDE's Eclipse y Netbeans son open source, es decir, de código abierto por lo que no tiene costo alguno, además tienen bastante soporte técnico como documentación, que es uno de los puntos más importante que se debe considerar al momento de elegir,

en especial si se trata de trabajar con el framework de Google GWT que también es de código abierto.

- En lo referente al tercer IDE IntelliJ IDEA, pues a pesar de ser licenciado, es decir, pagado y de más de gozar de popularidad para los autores descritos anteriormente, cuenta con el soporte técnico necesario para poder realizar cualquier aplicación web que deseamos detalle que es de mucha importancia para los desarrolladores.

3.3. Análisis de las herramientas

En la siguiente documentación se expresa en detalle todo lo necesario para explicar cada uno de las herramientas IDE's que se ha tomado como referencia anteriormente para el análisis comparativo.

3.3.1. Eclipse

3.3.1.1. Introducción

Gran parte de la programación de Eclipse fue realizada por IBM antes de que se creara el proyecto Eclipse como tal. El antecesor de Eclipse fue VisualAge y se construyó usando Smalltalk en un entorno de desarrollo llamado Envy. Con la aparición de Java en la década de los noventa, IBM desarrolló una máquina virtual válida tanto para Smalltalk como para Java. El rápido crecimiento de Java y sus ventajas con miras a una Internet en plena expansión obligaron a IBM a plantearse el abandono de esta máquina virtual dual y la construcción de una nueva plataforma basada en Java desde el principio. El producto final resultante fue Eclipse, que ya había costado unos 40 millones de dólares a IBM en el año 2001.

A finales de 2001 IBM, junto a Borland, crean la fundación sin fines de lucro Eclipse, abriéndose así al mundo de código abierto. A este consorcio se han unido poco a poco importantes empresas del desarrollo de software a escala mundial: Oracle, Rational Software, Red Hat, SuSE, HP, Serena, Ericsson y Novell, entre otras.

3.3.1.2. Definición

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse y que son usados también para desarrollar el mismo.

3.3.1.3. Arquitectura del IDE Eclipse

Un concepto fundamental de Eclipse, necesario para comprender lo que sigue, es el de recurso. En Eclipse, un recurso básico es simplemente un fichero ASCII que contiene código fuente para un lenguaje de programación.

El IDE Eclipse se compone de 6 grandes componentes:

a. El *Platform runtime*

Se encarga de gestionar los recursos y los plug-ins, además de permitir el arranque de la plataforma. Cuando se arranca Eclipse, este componente se encarga de buscar los ficheros de manifiestos de los plug-ins (que son archivos XML que describen los plug-ins), y carga

esta información en un registro. Solamente cuando se requiere por primera vez un plug-in, el *Platform runtime* lo ejecuta; este componente descubre de forma dinámica plug-ins durante el tiempo de ejecución. A grosso modo, el *Platform runtime* define los puntos de extensión y el modelo de plug-ins.

b. El *Workspace* (Espacio de trabajo)

Permite gestionar el acceso a ficheros tanto a alto como a bajo nivel. Actúa como un componente que encapsulara la gestión de archivos, permitiendo que los plug-ins utilicen sus métodos sin tener que trabajar directamente con distintos sistemas de archivos, según la plataforma que se utilice.

De la misma forma, refleja el estado actual de los proyectos locales, con su código fuente y sus ficheros compilados, que estén en la memoria activa. Al cerrar Eclipse se guarda el estado actual del *Workspace* local, de modo que cuando se reinicia Eclipse vuelve al estado en que se cerró.

c. El *Workbench* (Banco de trabajo)

Se encarga de la presentación de la información al usuario y de la gestión del diálogo con el mismo. Proporciona la interfaz gráfica de Eclipse y constituye uno de sus puntos más cuidados y atractivos. Desde el punto de vista del usuario una ventana del *Workbench* consiste en vistas y editores. Tanto la API como la implementación del *Workbench* se han realizado mediante SWT y JFace.

- **SWT (*Standard Widget Toolkit*)**

SWT es un conjunto de componentes para construir interfaces gráficas en Java, (widgets) desarrollados por el proyecto Eclipse; además recupera la idea original de la biblioteca AWT de utilizar componentes nativos, con lo que adopta un estilo más consistente en todas las plataformas, pero evita caer en las limitaciones de ésta.

- **JFACE**

JFace es un conjunto de herramientas de interfaz de usuario con las clases para el manejo de muchas tareas comunes de programación de interfaz de usuario. JFace es la ventana-independiente del sistema, tanto en su API y la aplicación, y está diseñado para trabajar con SWT.

El Workbench o banco de trabajo proporciona lo siguiente:

- **Editores**

Un editor es un componente que permite interaccionar con los contenidos de un fichero (no sólo con el código fuente, sino también con su fichero XML asociado, sus propiedades, etc.) y modificarlos.

- **Vistas**

Una vista proporciona metadatos sobre el recurso que se haya seleccionado: organización de un recurso dentro de un paquete o proyecto, estado de la compilación, etc.

- **Perspectivas**

Una perspectiva representa una configuración de editores y vistas relacionadas, al igual que opciones de menú y de compilación. Hay tres perspectivas estándar en Eclipse, pero usando plug-ins pueden añadirse otras nuevas. Al cambiar una perspectiva por otra se muestran diferentes editores, vistas y opciones de menú.

d. El componente de ayuda (*Help*)

Permite a los plug-ins proporcionar documentación HTML que pueda ser presentada contextualmente por el *Workbench*.

e. El componente de equipo (*Team o Team support*)

Se define un modelo de programación en equipo para crear y mantener un registro de las versiones de las aplicaciones que se desarrolle. Este componente permite que diferentes plug-ins de *repositorys* convivan dentro de la plataforma; del mismo modo, añade las vistas que el usuario necesite para interactuar con cualquier sistema de control de versiones que se esté usando. Tal y como se ha mencionado ya, Eclipse incluye de forma estándar un plug-in CVS, pero pueden añadirse otros *repositorys*, independientemente del VCM (*Version Control System*) que se utilice, la interfaz de usuario no cambia.

f. El componente de depuración (*Debug*)

Proporciona un modelo genérico de depuración, en el que se permite expresiones, puntos de interrupción, acciones habituales de depuración, etc., junto a una interfaz gráfica genérica

de depuración. Cualquier plug-in puede aprovechar los mecanismos de depuración que proporciona este componente.

Se debe mencionar que *Platform runtime* no es un plug-in; los otros cinco componentes son plug-ins, cualquier plug-in susceptible de incorporarse a Eclipse se integra con éste de la misma manera que cualquiera de los plug-ins que incluye de forma estándar. El *Workbench* y el *Workspace* son dos plug-ins indispensables porque proporcionan puntos de extensión usados por casi todos los plug-ins, pero su funcionamiento no difiere del de otros plug-ins.

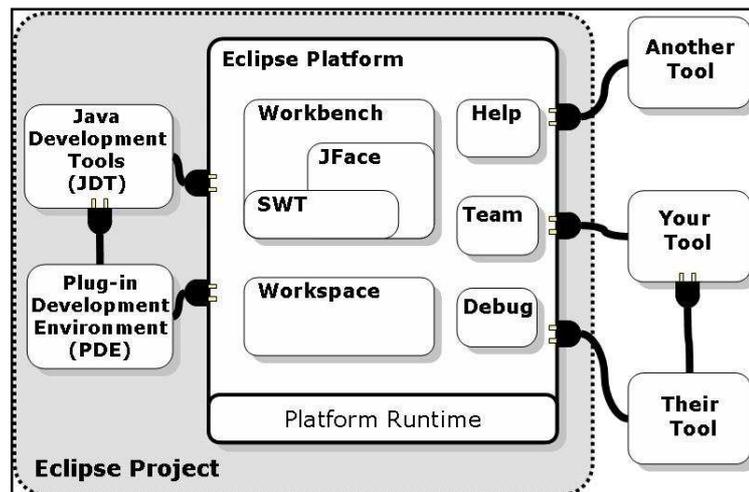


Grafico III.1: Arquitectura de la Plataforma Eclipse

3.3.1.4. Estructura del IDE Eclipse

La estructura del IDE de Eclipse está compuesta por la Máquina Virtual de Java y el SDK o Kit de Desarrollo Estándar de Eclipse que consta de tres elementos:

- La Plataforma de Eclipse
- La JDT (Herramienta de Desarrollo Java)

- El PDE (entorno de Desarrollo de Plu-ins)

a. La Plataforma de Eclipse

Es el núcleo básico o el kernel de Eclipse; emplea una estructura abierta de plug-ins (extensiones) que permite expandir las capacidades de la plataforma base; es de *arquitectura abierta*, por tanto es un producto de código fuente abierto u *open source*, está escrita en Java.

Provee a las capas superiores de servicios tales como editor de código fuente, infraestructura para depuración independiente del lenguaje de programación, soporte de versiones, búsqueda, compilación, asistentes para creación, etc.

b. La Herramienta de Desarrollo Java (JDT)

Agrupar un conjunto de plug-ins que extienden la plataforma básica proporcionando características de edición compilación, depuración y ejecución de código Java; además explica a la plataforma cómo entender Java, viene incluido en el SDK.

c. El Entorno de Desarrollo de Plug-ins (PDE)

Proporciona herramientas y asistentes que automatizan y facilitan la creación, desarrollo, depuración y distribución de plug-ins.

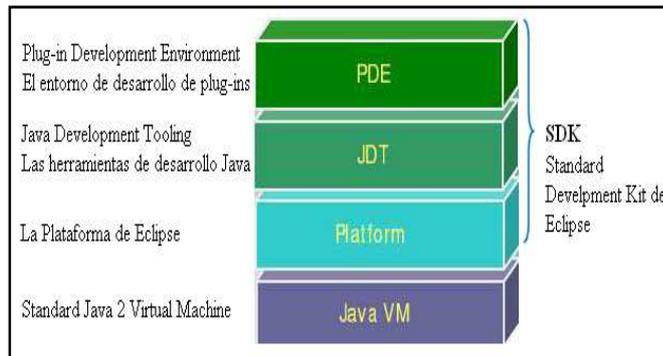


Grafico III.2: Estructura del IDE Eclipse

3.3.1.5. Características

La versión actual de Eclipse dispone de las siguientes características:

- Editor de texto
- Resaltado de sintaxis
- Compilación en tiempo real
- Pruebas unitarias con JUnit
- Control de versiones con CVS
- Integración con Ant
- Asistentes (*wizards*): para creación de proyectos, clases, tests, exportación e importación de proyectos; para generar esqueletos de códigos (templates), etc.
- Refactorización
- Multiplataforma (GNU/Linux, Solaris, Mac OSX, Windows)
- Soportado para distintas arquitecturas (x86, 64, ...)

Así mismo, a través de "plug-ins" libremente disponibles es posible añadir:

- Control de versiones con Subversion.

- Integración con Hibernate.

3.3.1.6. El proyecto Eclipse

El IDE Eclipse es, únicamente, una de las herramientas que se engloba bajo el denominado *Proyecto Eclipse*. El *Proyecto Eclipse* aún tanto el desarrollo del IDE Eclipse como de algunos de los plug-ins mas importantes (como el JDT, plug-in para el lenguaje Java, o el CDT, plug-in para el lenguaje C/C++).

Este proyecto también alcanza a las librerías que sirven como base para la construcción del IDE Eclipse (pero se puede utilizar de forma completamente independiente), como por ejemplo, la librería de widgets SWT.

3.3.1.7. Sub-proyectos de Eclipse

El proyecto global "Eclipse" se compone de diversos sub-proyectos:

- a. *Proyecto Eclipse*: orientado a la realización de herramientas de desarrollo altamente integrada.
- b. Proyecto de herramientas de Eclipse:
 - CDT: IDE para C/C++.
 - Graphical Editor Framework (GEF): para crear editores gráficos.
 - EMF: se crea herramientas y aplicaciones que se usa el modelo estructurado.
 - COBOL: IDE para COBOL.
 - Eclipse Visual Editor (VE): se crea constructores de GUIs para Eclipse, utilizando Swing, SWT, etc.

- UML2: para herramientas de modelización UML.
- c. Proyecto de herramientas web de Eclipse (WTP): orientado a la creación de herramientas de desarrollo web (editores, modelos, wizards, etc.), por ejemplo Web Standard Tools (WST), J2EE Standard Tools (JST), JavaServer Faces Tools (JSF), etc.
- d. Herramientas de monitorización, pruebas y análisis de rendimiento (Test & Performance Tools Platform, TPTP).
- e. Herramientas de Business Intelligence (BIRT).
- f. Data Tools Platform (DTP),
- g. Etc.

El proyecto Eclipse es el núcleo fundamental (por el momento). Sub-proyectos:

- a. La plataforma (rich client platform, RCP), un conjunto de frameworks y servicios comunes orientados a crear un marco global de herramientas.
- b. JDT (Java Development Tools). Aquí se centra el interés inmediato, ya que se orienta a la realización de IDEs (Integrated Development Environment), plug-ins, wizard, etc para el desarrollo Java.
- c. PDE (Plug-in Development Environment). Se centra en modelos y herramientas para el desarrollador de plug-ins.

3.3.1.8. La librería SWT

El entorno de desarrollo Eclipse, incluyendo sus plug-ins, está desarrollado por completo en el lenguaje Java. Un problema habitual en herramientas Java (como NetBeans) es que

son demasiado “pesadas”. Es decir, necesitan una máquina muy potente para poder ejecutarse de forma satisfactoria. En gran medida, estas necesidades vienen determinadas por el uso del API Swing para su interfaz gráfico.

Swing es una librería de widgets portable a cualquier plataforma que disponga de una máquina virtual Java pero a costa de no aprovechar las capacidades nativas del sistema donde se ejecuta, lo cual supone una ejecución sensiblemente más lenta que la de las aplicaciones nativas.

SWT es una librería de widgets equivalente a Swing en la cual, se aprovechan los widgets nativos del sistema sobre el que se ejecuta. El hecho de aprovechar los widgets nativos, permite que la ejecución de interfaces de usuario sea mucho más rápida y fluida que si se utilizase Swing y, además, siempre dispone del “Look and Feel” del sistema, sin necesidad de “emularlo”.

La contrapartida es que la librería SWT es nativa, es decir, es necesario disponer de una librería SWT específica para cada sistema operativo.

3.3.2. Netbeans

3.3.2.1. Introducción

Históricamente NetBeans es uno de los primeros IDEs que aparecieron para Java, y el primero de importancia que estaba escrito íntegramente en Java. Sin embargo una

compañía Checa estaba a punto de dar el gran salto con un IDE poco conocido (si se compara con los que existían antes), llamado NetBeans.

Sun Microsystems adquirió la compañía checa y designó a NetBeans como el IDE oficial para desarrollos Java, lo hizo gratuito y abrió su código. Sun sabía que si quería popularizar su joven plataforma y lenguaje, necesitaba disponer de una herramienta de desarrollo que fuese gratuita, y lógicamente, ésta tenía que estar desarrollada utilizando el propio Java para demostrar la capacidad y las posibilidades de la plataforma.

Este IDE tenía la ventaja de que fue creado por las mismas personas que crearon Java años antes y después de varios años de desarrollo ha llegado a ser tan útil y poderoso como Eclipse o quizás un poco más.

NetBeans es un entorno de desarrollo integrado (IDE por sus siglas en inglés). Esto quiere decir que integra todas las herramientas que necesitamos para poder desarrollar. Originalmente la programación en Java era algo complicada porque Java cuenta con una enorme cantidad de librerías y funciones que era preciso aprenderse de memoria, viendo esto muchas compañías construyeron diferentes entornos de programación para facilitar la tarea del programador.

3.3.2.2. Definición

NetBeans IDE es una aplicación de código abierto pensada para escribir, compilar, depurar y ejecutar programas para varios lenguajes, incluyendo a Java y C++ basado en la

plataforma NetBeans, con una serie de módulos específicos para un IDE. Tiene soporte para crear interfaces gráficas de forma visual, crear aplicaciones para móviles, desarrollar aplicaciones web y además estas funcionalidades son ampliables mediante instalación de paquetes adicionales. NetBeans IDE se conoce como la solución más completa para programar en Java.

NetBeans es también una **plataforma de ejecución** de aplicaciones, es decir, facilita la escritura de aplicaciones Java, proporcionando una serie de *servicios* comunes, que a su vez están disponibles a través del IDE.

3.3.2.3. Características

Entre las características que incluye este IDE tenemos las siguientes:

- Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).
- Contiene un sistema de proyectos basado en Ant, control de versiones y refactoring.
- Ofrece una serie de funcionalidades para los lenguajes de programación dinámicos Ruby y JavaScript, así como soporte para el último Java Standards.
- Incluyen soporte total para las nuevas tecnologías de sobremesa, como Beans Binding (JSR 295) y el entorno Swing Application Framework (JSR 296).
- Permite a los desarrolladores utilizar Ruby on Rails con el código Java existente.
- Mejor edición del código.

- Capacidades de navegación e inspección
- Historia local.
- Soporte integrado para subversión
- Mayores capacidades de personalización integradas en la distribución estándar.
- Facilidad de uso durante todo el ciclo de desarrollo
- Código abierto patrocinado por Sun
- Maneja la complejidad de la arquitectura orientada al servicio (SOA)
- El soporte al modelado mejora la productividad del desarrollador
- Promueve las mejores prácticas para la productividad del grupo
- Acelera el desarrollo.

Las nuevas versiones del IDE Netbeans incluye mejoras que se detallaremos a continuación:

PHP

- Editor de PHP (auto-completado de código, resaltado de código)
- Debug usando Xdebug
- Generadores de código para MySQL

JavaScript y Ajax

- Debug de JavaScript en Firefox y Internet Explorer
- Monitoreo de transacciones Http
- Editor JavaScript

- Administrador de librerías JavaScript, incluyendo Yahoo UI, Woodstock, JQuery, Prototype, Dojo y Scriptaculous
- Mejoras en el editor de CSS

Desarrollo Java EE y Web

- Mejoras en el soporte de Spring, Hibernate, JSF, generador de CRUD JSF, JPA (Java Persistence API)
- Generador de Web Services a partir de tablas de una base de datos o de entidades JPA
- Arrastrar-y-Tirar para servicios SaaS hacia archivos PHP
- Mejoras al editor SQL (auto-completado de código, guardar/recuperar consultas, y más)
- Mejoras en la importación y sincronización de proyectos Eclipse
- Auto-despliegue de la aplicación al guardar cambios

Groovy y Grails

- Desarrollo de aplicaciones Groovy puras, o utilización de Groovy dentro de proyectos Java SE
- Editor Groovy (auto-completado de código, resaltado y más)
- Framework de desarrollo de aplicaciones web Grails
- Abrir proyectos Grails existentes sin agregar metadata

Ruby y Rails

- Nueva interfaz del Test Runner para mostrar los resultados de las pruebas
- Mejoras en Rake: Rake Runner y archivos pre-generados de Rake
- Debugger: breakpoints condicionales y catchpoints
- Los proyectos Ruby aceptan opciones de la JVM de línea de comandos

GlassFish v3 Prelude para desarrollo web

- Arquitectura modular, basada en OSGi
- Bajo consumo de recursos, inicio y despliegue rápido
- Soporta scripting, incluyendo JRuby

C / C++

- Mejoras en el auto-completado de código, resaltado y mása
- Grafo de llamadas, ventana de memoria
- Desarrollo remoto
- Empaquetado de aplicaciones como archivos tar, zip o SVR4

Java ME

- Nuevo asistente para agregar componentes a la paleta del Diseñador Visual
- Nuevos componentes de UI SVG
- Nuevos Data Bindings para componentes personalizados
- Actualizada la herramienta de ofuscación a ProGuard 4.2 y el framework de pruebas a JUnit 1.1.0

Java Debugger

- Nuevo debugger para multi-hilos con mejoras en el UI y en el workflow
- Detector de deadlocks con indicadores visuales

Java SE

- Compilación al guardar automático
- Mejoras en la importación y sincronización de proyectos Eclipse
- Jerarquía de invocaciones Java
- Analizador de Javadoc

- El Swing UI Builder puede generar nombres de clases simples (en vez de nombres completos)

Herramientas del IDE y usabilidad

- Opciones de formateo por proyecto
- Un único shortcut de Búsqueda Rápida para las acciones del IDE, archivos, opciones y documentos
- Soporte para rutas de Windows UNC (carpetas de red compartidas).

3.3.2.4. Tipos de proyectos

Existen cuatro tipos de proyectos que se puede comenzar a crear en las versiones actuales del IDE Netbeans a partir de la 6.1:

- Module
- Library Wrapper Module
- Module Suite
- NetBeans Platform Application

Module (Módulo)

Un módulo es simplemente un típico JAR (Java ARchive) con cierta meta información almacenada en el manifiesto. Los módulos poseen la extensión NBM (NetBeans Module). Entre la meta información se encuentra la versión del módulo, las dependencias, descripción de funcionalidad, datos del autor, etc. Un módulo puede ser usado en cualquier proyecto desarrollado sobre NetBeans Platform (siempre que se cumplan las restricciones de dependencias del mismo), incluso sobre NetBeans IDE:

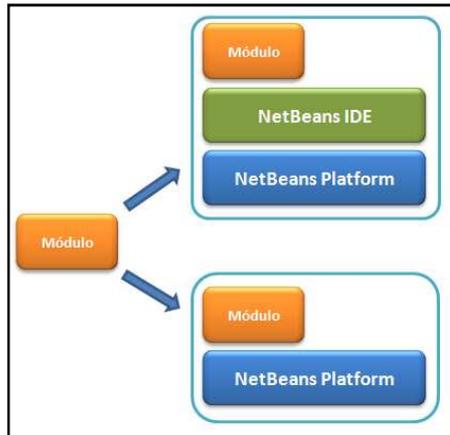


Grafico III.3: Módulos del IDE Netbeans

Library Wrapper Module (Módulo Envoltorio de Librería)

Un Library Wrapper Module permite que una librería externa (.JAR) sea vista desde los otros módulos como un módulo estándar. Básicamente se comporta igual que un Module, pero este no contendrá lógica alguna, simplemente expondrá la interfaz de la librería envuelta.



Grafico III.4: Módulo Envoltorio de Librería del IDE Netbeans

Tanto a los Module como a los Library Wrapper Module se los denomina de forma genérica como módulos o plug-ins.

Module Suite (Suite de Módulos)

Un Module Suite es una colección de módulos sobre la cual desarrollaremos nuestra aplicación. Por defecto, un Module Suite trae habilitados todos los módulos que conforman tanto a NetBeans Platform como a NetBeans IDE.

Clusters and Modules	Included
apisupport1	<input checked="" type="checkbox"/>
enterprise5	<input checked="" type="checkbox"/>
gsf1	<input checked="" type="checkbox"/>
harness	<input checked="" type="checkbox"/>
ide9	<input checked="" type="checkbox"/>
identity2	<input checked="" type="checkbox"/>
java2	<input checked="" type="checkbox"/>
nb6.1	<input checked="" type="checkbox"/>
platform8	<input checked="" type="checkbox"/>
profiler3	<input checked="" type="checkbox"/>
visualweb2	<input checked="" type="checkbox"/>
xml2	<input checked="" type="checkbox"/>

Grafico III.5: Módulo Suite del IDE Netbeans

El trabajo consistirá en deshabilitar los módulos que no se utilice y agregar los nuevos (Modules o Library Wrapper Module) para de esa forma crear la aplicación.

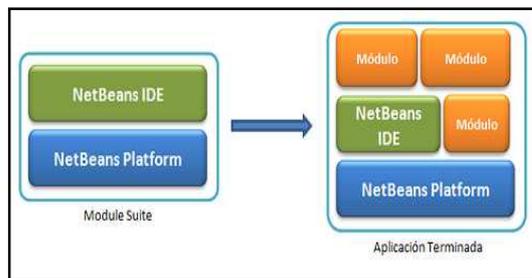


Grafico III.6: Deshabilitando los módulos en el IDE Netbeans

NetBeans Platform Application (Aplicación sobre NetBeans Platform)

Básicamente también crea un Module Suite, pero por defecto viene configurado con la mínima cantidad de módulos habilitados necesarios para iniciar. Por su configuración inicial lo llamaremos también como “Module Suite vacío”.

Clusters and Modules	Included
apisupport1	<input type="checkbox"/>
enterprise5	<input type="checkbox"/>
gsf1	<input type="checkbox"/>
harness	<input type="checkbox"/>
ide9	<input type="checkbox"/>
identity2	<input type="checkbox"/>
java2	<input type="checkbox"/>
nb6.1	<input type="checkbox"/>
platform8	<input checked="" type="checkbox"/>
profiler3	<input type="checkbox"/>
visualweb2	<input type="checkbox"/>
xml2	<input type="checkbox"/>

Grafico III.7: Aplicación sobre el IDE Netbeans

De forma contraria a un Module Suite, el trabajo consistirá en habilitar los módulos que se vaya requiriendo para que los módulos (ya sean Modules o Library Wrapper Module) cumplan sus dependencias, y así crear la aplicación.

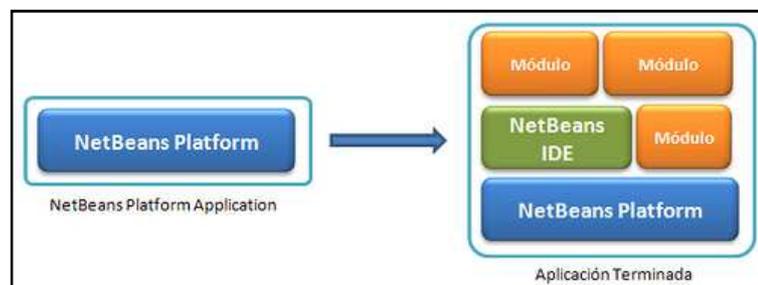


Grafico III.8: Habilitando los módulos del IDE Netbeans

3.3.2.5. Proceso de desarrollo

A continuación se mostrará los pasos correctos que se deben seguir para el proceso de desarrollo de un nuevo módulo en el IDE Netbeans.

- a. Crear el proyecto
- b. Configurar información general del modulo
- c. Utilizar plantillas para construir acciones, menús, ventanas, etc.
- d. Agregar la lógica al módulo
- e. Compilar, Debug y Prueba.
- f. Distribuir.

3.3.2.6. Ventajas

Entre las principales ventajas que se puede tener están las siguientes:

- La plataforma Netbeans RCP puede ser usada para desarrollar cualquier tipo de aplicación, no solamente plug-ins para Netbeans
- Reutilización de módulos
- Usando el Update Center module, se puede actualizar los módulos existentes de una aplicación o agregar nuevos.
- Los módulos permiten a las aplicaciones una abstracción muy alta y transparencia.
- El RCP ofrece muchas facilidades al desarrollador.
- En la última versión, se incluyen templates y Wizards que permiten desarrollar plug-ins de manera rápida y fácil.

3.3.2.7. Módulos Integrados de Netbeans

Los siguientes son módulos que forman parte de la base Netbeans IDE.

- **Profiler de NetBeans**

Es una herramienta para la optimización de los usos de Java: ayuda a encontrar los escapes de la memoria y a optimizar velocidad. Descargado antes por separado, se integra en la base IDE desde la versión 6.0.

Profiler se basa en un proyecto de investigación de los laboratorios del sol que fue nombrado JFluid. Esa investigación destapó las técnicas específicas que se pueden utilizar para bajar los gastos indirectos de perfilar un uso de Java. Una de esas técnicas es la instrumentación dinámica del bytecode, que es particularmente útil para perfilar los usos grandes de Java.

Usando la instrumentación dinámica del bytecode y algoritmos adicionales, el Profiler de NetBeans puede obtener la información runtime sobre los usos que son demasiados grandes o complejo para otros profilers. NetBeans también apoya perfilar los puntos que le dejan perfilar puntos exactos del tiempo de la ejecución y de ejecución de la medida.

- **Herramienta de diseño del GUI**

Conocido antes como *proyecto Matisse*, la diseñar-herramienta del GUI permite al prototipo y al diseño Oscilación GUIs arrastrando y colocando componentes del GUI.

El constructor del GUI tiene también ayuda incorporada para JSR 296 (Marco del uso del oscilación), y JSR 295 (habas que atan tecnología).

- **Redactor del Javascript de NetBeans**

El redactor del Javascript de NetBeans proporciona la ayuda extendida para Javascript y CSS. Las características abarcan:

- a. Redactor del Javascript:**

- El destacar del sintaxis
- El refactoring
- Cifre la terminación para los objetos y las funciones nativos
- Características de todo el redactor de NetBeans
- generación del esqueleto de la clase del Javascript
- Generación de AJAX servicio repetido de una plantilla
- Compatibilidad del browser cheque;

- b. Redactor de CSS:**

- Cifre la terminación para los nombres de los estilos
- Navegación rápida a través del panel del navegador
- Exhiba el declaración de la regla del CSS en una opinión de la lista
- Estructura de archivo de exhibición en una opinión del árbol
- Clasifique la opinión del contorno orden por nombre, del tipo o del declaración (lista y el árbol)
- Cree el declaración de la regla (el árbol solamente),
- Refactor a la parte de un nombre de la regla (árbol solamente).

3.3.2.8. Paquetes de Netbeans IDE

Los usuarios pueden elegir instalar los paquetes de NetBeans IDE al instalar el IDE, en una base de uno a uno.

- **Paquete de la movilidad de NetBeans**

Es una herramienta para desarrollar los usos que funcionan en los dispositivos móviles; los teléfonos generalmente móviles, pero éste también incluye el entrada-nivel PDAs, entre otros.

El paquete de la movilidad se puede utilizar para escribir, para probar, y para eliminar errores de los usos para: Edición micro de Java la plataforma (Java YO plataforma) tecnología que permitió los dispositivos móviles. Integra la ayuda para Perfil móvil del dispositivo de información (MIDP) 2.0, Configuración de dispositivo limitada conectada (CLDC) 1.1, y Configuración de dispositivo conectada (CDC). Se puede integrar fácilmente los emuladores de tercera persona para un ambiente de prueba robusto. Actualmente, el paquete está disponible como dos racimos separados, una CDC que cubre y la otra, dispositivos de CLDC.

- **Paquete de NetBeans C/C++**

Agrega la ayuda para los reveladores de C/C++ a NetBeans IDE 5.5. Este paquete deja a reveladores de C/C++ utilizar su sistema especificado de recopiladores y de

herramientas conjuntamente con NetBeans IDE para construir los usos nativos para MS Windows, Linux, y Solaris.

El paquete hace a redactor lengua-enterado para C/C++, y proporciona las plantillas del proyecto, un browser dinámico de la clase, hace la ayuda del archivo, y la funcionalidad de la depuración. Los reveladores pueden también ampliar el paquete con sus propias funcionalidades adicionales.

- **Paquete de la empresa de NetBeans**

Amplía la ayuda para el desarrollo del uso de la empresa y servicios del Web en NetBeans IDE 5.5.

El paquete de la empresa amplía las capacidades del IDE para escribir, para probar y para eliminar errores de arquitectura servicio-orientada (SOA) usos usando XML, BPEL, y servicios de la tela de Java. El paquete agrega las herramientas de diseño visuales para UML que modela, esquema de XML, y orchestration de los servicios de la tela, y ayuda para los servicios identidad-permitidos seguros de la tela.

También instala y configura los tiempos de pasada necesarios, incluyendo un servidor de la gerencia del motor y de la identidad de BPEL integrado con el servidor del uso del sistema de Java del sol.

- **Paquete visual del Web de NetBeans**

Proporciona un interfaz fácil e intuitivo de la fricción y de la gota para los reveladores crear UIs rico para el desarrollo de la tela. Estos componentes se basan encendido JSF y ayuda en el desarrollo fácil para los interfaces utilizador. Este paquete visual de la tela se considera ser mucho mejor y versión mejorada con muchos de buenas características comparadas a los lanzamientos anteriores del creador del estudio de Java del sol.

- **Paquete de rubíes de NetBeans**

Desde la versión 6.0, Netbeans permite el desarrollo del IDE con Rubí y JRuby, así como Carriles para estas dos puestas en práctica del rubí.

El paquete de rubíes incluye funcionalidades del redactor como:

- El corregir básico,
- Sintaxis que destaca para el rubí
- Cifra la terminación
- El destacar de la ocurrencia,
- La documentación integrada hacer estallar-sube para las llamadas del API del rubí,
- Análisis semántico con destacar de parámetros y de variables locales inusitadas,
- El eliminar errores de código de rubíes.

3.3.3. INTELLIJ IDEA

3.3.3.1. Introducción

La primera versión de IntelliJ IDEA apareció en enero de 2001, y en ese momento era el único IDE de Java disponible con el código de navegación avanzada y el código de refactorización capacidades integradas.

3.3.3.2. Definición

IntelliJ IDEA se autodefine como un entorno inteligente para desarrollar aplicaciones Java, cliente y servidor. Efectivamente, se trata de un completísimo IDE que también permite desarrollar aplicaciones para móviles.

IntelliJ IDEA posee un avanzado editor de código, compatible con multitud de tecnologías (AJAX, JSP, EJB...) y, dentro de un mismo entorno, ofrece análisis del código, compilación/ejecución/debugging, control de versiones, detección de duplicaciones, análisis de dependencias y soporte para plug-ins.

3.3.3.3. Características

La última versión de IntelliJ IDEA (9,0) incluye lo siguiente:

- Diagramas de clases UML
- Modelado Visual Hibernate
- Spring 3,0 apoyo
- Dependencia y análisis de flujo de datos

- Apache Maven apoyo
- Refactorizaciones de lenguajes múltiples
- Código de Inspecciones y Revisiones
- Contrato Anotaciones
- Código de detección de duplicados
- Análisis de dependencia.

3.3.3.4. Ventajas

Entre las principales ventajas que tiene este IDE tenemos:

- Permite a los desarrolladores centrarse en el desarrollo y se ocupa de todas las tareas de rutina.
- Permite escribir, depurar, refactorizar, probar y aprender el código sin ningún problema.
- Manejo de código mixto en base a Java, Ruby, Groovy, Python, y Scala.
- Mantiene automáticamente la calidad del código.
- Las pistas y los errores se fija en todos los niveles - desde las declaraciones a la general arquitectura.
- Desarrollo de código, a menos tiempo.
- Diseñado para trabajar en proyectos de toda escala - de individuo a nivel empresarial.
- Compatible con los principales lenguajes, las tecnologías y los marcos.
- Funciona con las versiones sistemas de control y TeamCity, el servidor de integración continua.

- Nuevo soporte para SQL, que permite editar guiones de base de datos y ejecutar consultas directamente en el IDE, con una amplia gama de funciones de asistencia de codificación para diversos dialectos SQL.

Una significativa mejora en el rendimiento en todo el producto, incluida una reducción de los tiempos de inicio, armados de proyecto más rápido, y actualizaciones de control de versiones

3.3.3.5. Función

Se han añadido muchas funciones esenciales de Java, incluidas 7 nuevas refactorizaciones, muchas inspecciones de código nuevas, nuevas herramientas para trabajar con pruebas de unidades, funciones innovadoras tales como una finalización de códigos aún más inteligente, y la función "Dataflow to This", y mucho más

3.4. Determinación de los parámetros de comparación

Los parámetros de comparación para un análisis comparativo son de mucha importancia pues son los que permiten valorar en este caso el IDE más adecuado o idóneo que se puede tener para el desarrollo de una aplicación usando la plataforma RIA-GWT, y para el propósito del presente análisis se ha considerado los parámetros de Acceso a datos, Compilación, Interfaz Gráfica de Usuario (GUI), Seguridad y Soporte Técnico de GWT, por ser los que más se adaptan a las características y funcionalidades de los IDE's utilizados para el presente análisis comparativo.

3.4.1. Parámetro 1: Acceso a la Base de Datos

En el presente parámetro trata sobre las características que se puede tener en GWT para el acceso a la base de datos, las mismas que se pueden evaluar dependiendo de las utilidades y formas que se tengan para realizar esta tarea, además de la compatibilidad del driver con el IDE y el motor de base de datos DB2 que se esté evaluando.

3.4.2. Parámetro 2: Compilación

En este parámetro se analizará y comparará la forma en que se realiza la compilación de una aplicación web realizada con GWT, evaluando los tiempos de respuesta que se demora para la depuración considerando el número de líneas de código, la capacidad para detectar errores y el número de pasos que se requiere para el acceso a la base de datos.

3.4.3. Parámetro 3: Interfaz Gráfica de Usuario (GUI)

Este parámetro es el que permitirá saber la facilidad que se tiene en GWT para el desarrollo de la Interfaz Gráfica de Usuario (GUI) y la reusabilidad que pueda tener la misma dentro de la aplicación.

3.4.4. Parámetro 4: Seguridad

En este parámetro se analizará la seguridad que provee la plataforma GWT para el desarrollo de una aplicación web, basándose en los protocolos de seguridad que soportan, el encapsulamiento del código puro de java, el encapsulamiento del código HTML, así como, la consistencia de las claves para el ingreso al sistema para al final saber si ofrece la

protección necesaria contra amenazas para reducir riesgos, garantizar la integridad, confidencialidad y disponibilidad.

3.4.5. Parámetro 5: Soporte Técnico de GWT

En este módulo se analizará y comparará la documentación y las condiciones de uso que tenga GWT con los distintos entornos de desarrollo integrados (IDE). En relación a documentación se tomará en cuenta la información publicada en los sitios oficiales de GWT así como los respectivos proyectos realizados, y que IDE se ha usado para el desarrollo de los mismos.

3.5. Determinación de las variables para los parámetros de comparación

3.5.1. Parámetro 1: Acceso a la Base de Datos

- Utilización de RPC
- Número de Plug-in para el acceso a la base
- Grado de dificultad para el acceso a la base de datos
- Número de pasos que se necesita para el acceso

3.5.2. Parámetro 2: Compilación

- Número de líneas de código.
- Tiempo mínimo de depuración.
- Capacidad para detectar errores.
- Capacidad para depurar errores.

3.5.3. Parámetro 3: Interfaz Gráfica de Usuario (GUI)

- Diseñador visual para GUI
- Generación de métodos set y get
- Menor coste
- Reutilización de componentes gráficos

3.5.4. Parámetro 4: Seguridad

- Encapsulamiento del código JAVA
- Encapsulamiento de la página HTML
- Números de Protocolos
- Consistencia de las claves para el acceso al sistema

3.5.5. Parámetro 5: Soporte Técnico de GWT

- Ayuda en línea del sitio oficial de GWT
- Páginas de ayuda de GWT
- Licencias

3.6. Análisis comparativo

En esta sección se va mostrar el estudio de GWT en los siguientes Entornos Integrado de Desarrollo (IDE) Eclipse, Netbeans, y INTELLIJ IDEA los cuales se usan para desarrollar aplicaciones web usando java, la comparación se realizará por medio de cuadros comparativos en los se va a determinar las variables de cada uno de los parámetros tomados

en cuenta para el respectivo análisis, seguidos de una valoración, interpretación y calificación del criterio evaluado por parte de los autores.

Para obtener resultados cuantitativos y cualitativos que permitan una selección adecuada de uno de los IDE's analizados para trabajar con GWT se realizará la calificación de cada uno de los parámetros de comparación que se basa en la siguiente escala:

Tabla III.2: Escala de puntuación para calificación de parámetros

Regular	Bueno	Muy Bueno	Excelente
< 70%	>= 70% y <80%	>= 80% y <90%	>= 90%

La siguiente tabla permitirá saber la valoración cuantitativa para de las variables de los parámetros seleccionados:

Tabla III.3: Escala de valoración cuantitativa para los parámetros

1	2	3
No se cumple	Se cumple parcialmente	Se cumple
Malo	Bueno	Excelente
Deficiente	Poco eficiente	Muy eficiente
Nada	Poco	Mucho
No	Más o menos	Si

Donde cada una de las variables va a ser evaluada sobre el máximo que es 3 y cada uno de los ítems de la interpretación incluye la siguiente nomenclatura (x,y,z) / w en donde cada letra significa lo siguiente:

x: Representa el puntaje que obtiene el IDE Eclipse.

y: Representa el puntaje que obtiene el IDE Netbeans.

z: Representa el puntaje que obtiene el IDE INTELLIJ IDEA.

w: Representa la base del puntaje sobre la cual se está calificando el parámetro.

La calificación definitiva de la herramienta en base a cada parámetro de comparación se obtiene sumando los puntajes obtenidos del análisis, utilizando las siguientes formulas:

$$\mathbf{PEclip= \Sigma(x), PNetb= \Sigma(y), PIntel= \Sigma(z), Pc= \Sigma(w)}$$

$$\mathbf{Calificación de Eclipse (Cc - Ecli) = (PEclip/Pc)*100\%}$$

$$\mathbf{Calificación de Netbeans (Cc - Netb) = (PNetb/Pc)*100\%}$$

$$\mathbf{Calificación de IntelliJ IDEA (Cc - Intel) = (PIntel/Pc)*100\%}$$

En donde:

PEcli: Puntaje acumulado por Eclipse en el parámetro analizado

PNetb: Puntaje acumulado por Netbeans en el parámetro analizado

PIntel: Puntaje acumulado por IntelliJ IDEA en el parámetro analizado

Pc: Puntaje sobre el que se califica el parámetro analizado

Cc – Ecli: Porcentaje de la calificación total que obtuvo Eclipse en el parámetro

Cc – Netb: Porcentaje de la calificación total que obtuvo Netbeans en el parámetro

Cc – Intel: Porcentaje de la calificación total que obtuvo IntelliJ IDEA en el parámetro

3.6.1. Parámetro 1: Acceso a la Base de Datos

3.6.1.1. Determinación de Variables

- a. Utilización de RPC
- b. Número de plug-in para el acceso a la base
- c. Grado de dificultad para el acceso a la base de datos
- d. Número de pasos que se necesita para el acceso

3.6.1.2. Valoraciones

a. Variable Utilización de RPC

La utilización de las llamadas a procedimientos remotos (RPC) cuando se usa la plataforma de desarrollo GWT es uno de los factores más importantes si se trata de extraer datos de la base, con esta variable se evaluará si el IDE soporta o no el RPC y el grado de complejidad que pueda tener.

b. Variable Número de plug-in para el acceso a la base

El presente indicador evaluará la cantidad de plug-in que necesita GWT para poder trabajar perfectamente con la base de datos permitiendo también la creación de los distintos módulos que se puedan ocupar en el IDE seleccionado.

c. Variable Grado de dificultad para el acceso a la base de datos

El grado de dificultad que se pueda tener para poder acceder a la base de datos es una variable muy importante para el desarrollo de una aplicación web empresarial, pues es la que nos indicará si se puede o no trabajar con la base usando GWT definiendo así su usabilidad.

d. Variable Número de pasos que se necesita para el acceso

Este indicador permitirá saber el número de pasos mínimos que se requiere para acceder a la base usando GWT, basándose en uno de los IDE que se ha seleccionado para el análisis comparativo.

Tabla III.4: Variable Acceso a la Base de Datos

Variable	Eclipse	Netbeans	Intelij IDEA
Utilización de RPC	Excelente	Excelente	Poco eficiente
Número de plug-in para el acceso a la base	Lo necesario	Muchos	Muchos
Grado de dificultad para el acceso	Mucho	Mucho	Mucho
Número de pasos que se necesita para el acceso	Poco	Nada	Nada

3.6.1.3. Interpretación

- La utilización de RPC para el acceso a la base de datos es de mucha importancia para cualquier aplicación sea esta de escritorio o web, por lo que se puede decir que tanto eclipse como netbeans tienen la utilización adecuada para RPC usando GWT, mientras que cuando se usa IntelliJ IDEA con GWT el RPC es poco eficiente. (3,3,2)/3
- El número de plug-in que se pueden usar para el acceso a la base de datos en cada uno de los IDE's analizados usando la tecnología GWT es importante para cualquier aplicación, pues de ellos depende que la aplicación sea rápida o no sabiendo que mientras menos plug-in se tenga más eficiente es la aplicación. Por tanto, eclipse emplea los plug-in necesarios para la realización de una aplicación mientras que los

dos IDE's restantes utilizan demasiados plug-in innecesarios para el empleo de la plataforma GWT. (2,3,3)/3

- En lo que se refiere al grado de dificultad que se puede tener para el acceso a la base usando GWT tenemos que todos los IDE's tienen el mismo grado, es decir, que resulta difícil el aprendizaje de los RPC. (3,3,3)/3
- En lo referente al número de pasos que se necesitan para lograr el acceso a la base teniendo instalada la plataforma GWT, los IDE's netbeans e IntelliJ IDEA no usan la misma cantidad de pasos por lo que ahorran tiempo, mientras que eclipse si usa una secuencia ordenada de pasos contrario a lo anterior. (2,1,1)/3

3.6.1.4. Calificación

$$P_c = \sum(w) = 3+3+3+3 = 12$$

$$PE_{\text{Ecli}} = \sum(z) = 3+2+3+2 = 10 \quad C_c - \text{Ecli}: PE_{\text{Ecli}} / P_c = (10/12) * 100\% = 83,33\%$$

$$P_{\text{Netb}} = \sum(x) = 3+3+3+1 = 10 \quad C_c - \text{Netb}: P_{\text{netb}} / P_c = (10/12) * 100\% = 83,33\%$$

$$P_{\text{Intel}} = \sum(y) = 2+3+3+1 = 9 \quad C_c - \text{Intel}: P_{\text{Intel}} / P_c = (9/12) * 100\% = 75\%$$

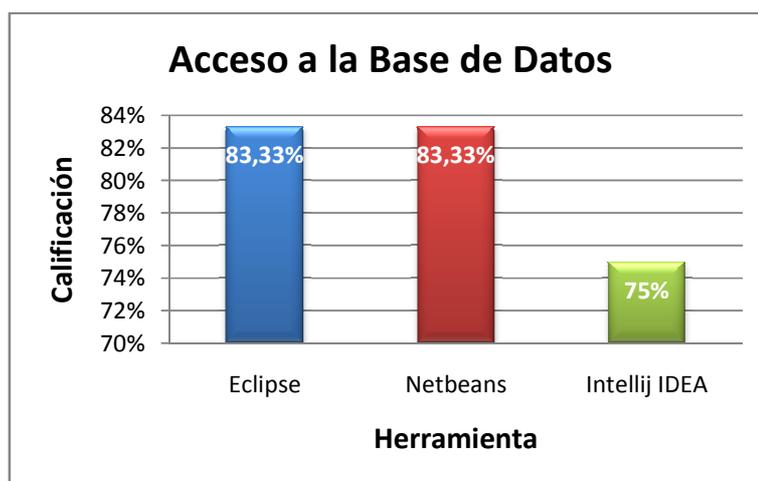


Gráfico III.9: Comparación Estadística del parámetro de Acceso a la Base de Datos

3.6.2. Parámetro 2: Compilación

3.6.2.1. Determinación de Variables

- a. Número de líneas de código
- b. Tiempo mínimo de depuración
- c. Capacidad para detectar errores
- d. Capacidad para depurar errores

3.6.2.2. Valoraciones

a. Variable Número de líneas de código

Este indicador permitirá evaluar la cantidad de líneas de código que se llevan para realizar un determinado proceso en el GWT con la finalidad de establecer el IDE que menos líneas ocupa con el uso de esta plataforma.

b. Variable Tiempo mínimo de depuración

La variable tiempo mínimo de depuración permitirá saber cuál de los IDEs seleccionados para el análisis nos dará el menor tiempo para mejorar la aplicación web usando la plataforma de desarrollo GWT.

c. Variable Capacidad para detectar errores

En esta variable se indicará la capacidad que se puede tener en cada IDE usando GWT para detectar la mayor cantidad de errores de una aplicación.

d. Variable Capacidad para depurar errores

Esta variable evaluará el nivel de confiabilidad que se pueda tener para poder depurar un error encontrado en la aplicación, mostrando el lugar exacto en el que se encuentra

sirviendo así de apoyo para el programador para que de esta manera el usuario llegue al cumplimiento de sus objetivos.

Tabla III.5: Variable Compilación

Variable	Eclipse	Netbeans	IntelliJ IDEA
Número de líneas de código	Mucho	Poco	Mucho
Tiempo mínimo de depuración	Poco	Mucho	Mucho
Capacidad para detectar errores	Eficiente	Poco eficiente	Poco Eficiente
Capacidad para depurar errores	Eficiente	Eficiente	Poco Eficiente

3.6.2.3. Interpretación

- El número mínimo de líneas de código que se puede tener en una aplicación quiere decir que será muy eficiente, por lo que después de haberse realizado la respectiva representación en la tabla anterior, se puede determinar que de los tres IDE's analizados para el uso de GWT el netbeans es el que menos líneas utiliza obteniendo así la siguiente calificación. (2,3,2)/3
- En lo que se refiere al tiempo de depuración que se puede tardar cada IDE mientras utilizad GWT, se ve claramente como eclipse es que menos tiempo utiliza en relación con los demás IDE's por los que se les da la siguiente calificación (3,2,2)/3
- Para un desarrollador una de las cosas importantes que debe considerar es que para detectar errores lo primordial sería que esta detección sea lo más eficiente posible, y en cuanto al uso de la plataforma GWT con un determinado IDE que se tiene para el análisis podemos ver que eclipse ofrece mejores prestaciones para esta variable (3,2,2)/3

- Así como en el punto anterior la detección de errores es importante la depuración tiene el mismo peso y eclipse sigue siendo eficiente y junto a netbeans llevan la ventaja sobre IntelliJ IDEA, obteniendo su respectiva calificación. (3,3,2)/3

3.6.2.4. Calificación

$$P_c = \sum(w) = 3+3+3+3 = 12$$

$$PE_{\text{Ecli}} = \sum(z) = 2+3+3+3 = 11$$

$$P_{\text{Netb}} = \sum(x) = 3+2+2+3 = 10$$

$$P_{\text{Intel}} = \sum(y) = 2+2+2+2 = 8$$

$$C_c - \text{Ecli}: PE_{\text{Ecli}} / P_c = (11/12) * 100\% = 91,67\%$$

$$C_c - \text{Netb}: P_{\text{Netb}} / P_c = (10/12) * 100\% = 83,33\%$$

$$C_c - \text{Intel}: P_{\text{Intel}} / P_c = (8/12) * 100\% = 66,67\%$$

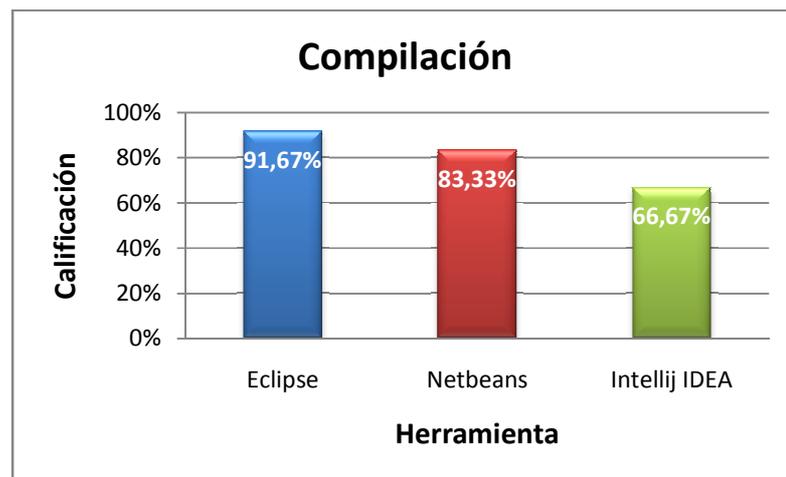


Gráfico III.10: Comparación Estadística del parámetro de la Compilación

3.6.3. Parámetro 3: Interfaz Gráfica de Usuario (GUI)

3.6.3.1. Determinación de Variables

- a. Diseñador visual para GUI
- b. Generación de métodos set y get
- c. Coste

d. Reutilización de componentes gráficos

3.6.3.2. Valoraciones

a. Variable Diseñador visual para GUI

Con este indicador se valorará si cuando se usa GWT el IDE es capaz de generar el diseñador gráfico para de esta manera facilitar al usuario la tarea de poder diseñar y programar su aplicación.

b. Variable Generación de métodos set y get

Este indicador permitirá saber si con el uso de GWT se puede o no generar los métodos set y get los cuales nos permitirán darle valor a cualquier variable y de la misma manera permitirá obtener el valor de la variable que necesitamos con la ayuda del método get.

c. Variable Menor coste

Al momento de desarrollar una aplicación web empresarial se piensa rápidamente en el coste que esta nos puede representar, con este indicador se dará una visión más exacta sobre cuál de los IDE's seleccionados representa un menor coste para poder usar GWT con un GUI, considerando siempre las herramientas que sean de código libre.

d. Variable Reutilización de componentes gráficos

Este indicador evaluará los aspectos de reutilización de componentes gráficos que pueda tener con GWT en un IDE, teniendo en cuenta que GWT usa los widget primitivos lo que ayuda a utilizar demasiados recursos en la máquina y poder tener menor tiempo de compilación.

Tabla III.6: Variable Interfaz Gráfica de Usuario (GUI)

Variable	Eclipse	Netbeans	Intelij IDEA
Diseñador visual para GUI	No cumple	No cumple	No cumple
Generación de métodos set y get	Se cumple	Se cumple	No cumple
Menor coste	Si	Si	No
Reutilización de componentes gráficos	Si	Más o menos	No

3.6.3.3. Interpretación

- Después de lo analizado anteriormente se puede decir que GWT no permite tener en ninguno de los IDE's analizados un diseñador visual para GUI gratis, por lo que van a tener la siguiente puntuación (1,1,1)/3
- La generación de los métodos set y get durante el desarrollo de la aplicación pueden resultar muy importantes y más si se trata analizar el uso de la plataforma GWT, por ello se ve que tanto eclipse como Netbeans cumplen con esta característica mientras que el IntelliJ IDEA queda en un segundo plano.(3,3,1)/3
- En relación al coste se puede decir que si bien es cierto GWT no tiene ningún valor.es importante que el IDE ene el que se ,o va a istalar o usar tampoco lo tenga, por tanto, se ve en la tabla que tanto eclipse como Netbeans son gratuitos, cosa que no ocurre con IntelliJ IDEA que es licenciado teniendo obiamente un puntaje menor al los demás (3,3,1)/3
- Para todo programador es importante poder contar con un componente o plataforma que permita la reutilización de componentes gráficos el mismo que ayudará a realizar un diseño de interfaz mucho más rápido y agradable, por lo que eclipse permite hacer

esta tarea gracias al uso de widget primitivos, y los otros dos IDE's tratan pero no llegan a tener la misma ventaja que eclipse. (3,2,1)/3

3.6.3.4. Calificación

$$P_c = \sum(w) = 3+3+3+3 = 12$$

$$PE_{\text{Ecli}} = \sum(x) = 1+3+3+3 = 10$$

$$C_c - \text{Ecli}: PE_{\text{Ecli}} / P_c = (10/12) * 100\% = 83,33\%$$

$$P_{\text{Netb}} = \sum(y) = 1+3+3+2 = 9$$

$$C_c - \text{Netb}: P_{\text{Netb}} / P_c = (9/12) * 100\% = 75\%$$

$$P_{\text{Intel}} = \sum(z) = 1+1+1+1 = 4$$

$$C_c - \text{Intel}: P_{\text{Intel}} / P_c = (4/12) * 100\% = 33,33\%$$

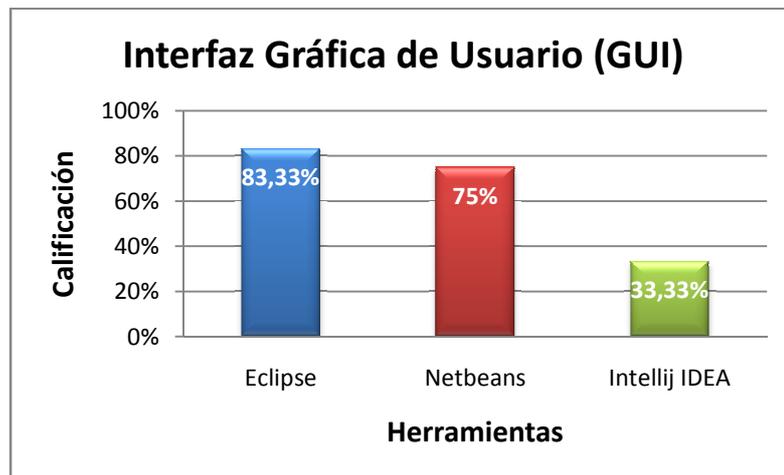


Gráfico III.11: Comparación estadística del parámetro Interfaz Gráfico de Usuario (GUI)

3.6.4. Parámetro 4: Seguridad

3.6.4.1. Determinación de Variables

- Encapsulamiento del código java
- Encapsulamiento de la página html
- Número de protocolos
- Consistencia de las claves para el acceso al sistema.

3.6.4.2. Valoraciones

a. Variable Encapsulamiento del código JAVA

Mediante este indicador se evaluará la capacidad que pueda tener GWT en un IDE para encapsular el código java con el que se ha realizado la aplicación web.

b. Variable Encapsulamiento de la página HTML

Mediante esta variable se valorará la capacidad que pueda tener GWT en un IDE para encapsular el código HTML con el que se realizó el diseño de la aplicación web.

c. Variable Números de protocolos

Este indicador nos permitirá evaluar los protocolos más seguros que podemos encontrar en la transmisión de la información usando GWT en un IDE determinado.

d. Variable Consistencia de las claves para el acceso al sistema

Con esa variable consistencia de las claves para el acceso al sistema, GWT se asegura que los datos ingresados para inicio de sesión sean los correctos y concuerden con los que se encuentra en la base de datos, y arrojarles un mensaje de error si sucede lo contrario.

Tabla III.7: Variable Seguridad

Variable	Eclipse	Netbeans	Intelij IDEA
Encapsulamiento del código Java	Se cumple	Se cumple	Se cumple
Encapsulamiento de la página html	Se cumple	Se cumple	Se cumple
Números de protocolos	Excelente	Excelente	Excelente
Consistencia de las claves para el acceso al sistema	Se cumple	Se cumple	Se cumple

3.6.4.3. Interpretación

- Con respecto con el encapsulamiento del código Java o código fuente que ofrece GWT en cada uno de los IDE's analizados se puede decir que en todos cumple a cabalidad por los que se encuentran empatados en cuanto a la calificación (3,3,3)/3
- Igual que el encapsulamiento anterior GWT también ofrece esta seguridad al esconder la programación del código HTML sin importar el IDE que se use, tal como se lo demostró en la tabla anterior teniendo una igualdad de puntos (3,3,3)/3
- El número de protocolos que usa GWT es el mismo para cualquier IDE que se use, ocupando siempre los necesarios por tanto llegana una puntuación de igualdad. (3,3,3)/3
- En lo que se refiere a la consistencia que se debe tener con las claves de ingreso al sistema, GWT ofrece la posibilidad de validar los datos ingresados por el usuario obteniendo resultados esperados, y en la tabla se puede observar que en los IDE que se analice se va a tener el mismo resultado (3,3,3)/3

3.6.4.4. Calificación

$$Pc = \sum(w) = 3+3+3+3 = 12$$

$$PEcli = \sum(x) = 3+3+3+3 = 12$$

$$PNetb = \sum(y) = 3+3+3+3 = 12$$

$$PIntel = \sum(z) = 3+3+3+3 = 12$$

$$Cc - Ecli: PEcli / Pc = (12/12)*100\% = 100\%$$

$$Cc - Netb: PNetb / Pc = (12/12)*100\% = 100\%$$

$$Cc - Intel: PIntel / Pc = (12/12)*100\% = 100\%$$

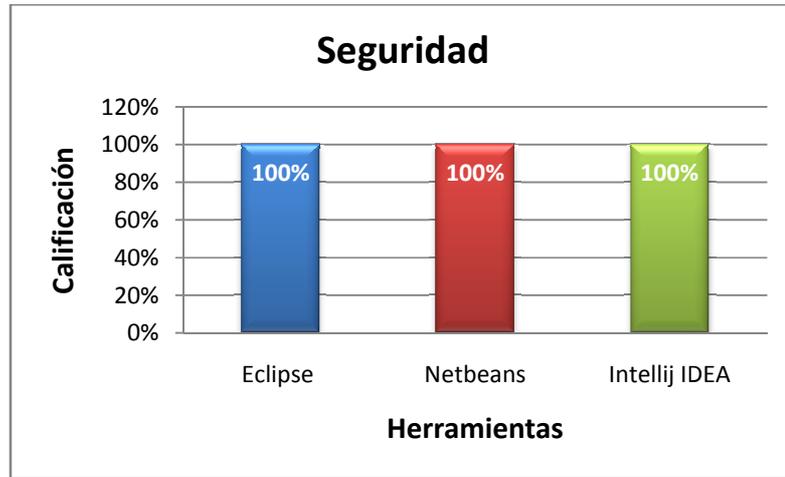


Gráfico III.12: Comparación estadística del parámetro de la seguridad

3.6.5. Parámetro 5: Soporte Técnico de GWT

3.6.5.1. Determinación de Variables

- a. Ayuda del sitio oficial de GWT
- b. Páginas de ayuda de GWT
- c. Licencias

3.6.5.2. Valoraciones

a. Variable Ayuda del sitio oficial de GWT

Este indicador se basa en la calidad de la información publicada en sitios web oficiales de GWT para ser usado en cualquier IDE principalmente los que se tomarán en cuenta para el análisis.

b. Variable Páginas de ayuda de GWT

Para el empleo de la plataforma RIA-GWT la mayoría de los IDE's pueden usar páginas que contienen información necesaria para su realización, por esta razón se evaluará la

existencia de esta utilidad y obtener el mejor desarrollo gracias a la información brindada por la web.

c. Variable Licencias

Este indicador evaluará que la licencia de GWT con las que gozan cada uno de los entornos de desarrollo integrado (IDE) que se van a comparar sean código libre tomando en cuenta los aspectos relacionados con el desarrollo del presente proyecto.

Tabla III.8 Variable Soporte Técnico con respecto a GWT

Variable	Eclipse	Netbeans	Intellij IDEA
Ayuda del sitio oficial de GWT	Se cumple	Poco	No cumple
Páginas de ayuda de GWT	Se cumple	Poco	Nada
Licencias	Gratis	Gratis	Licenciada

3.6.5.3. Interpretación

- En lo relacionado a la ayuda en línea que brinda GWT con respecto a los IDE's analizados, tenemos que según la web eclipse goza en su plenitud de esto, pues toda la información se habla con eclipse, un poco con netbeans y nada con el IntelliJ IDEA.
(3,2,1)/3
- En cuanto a la información que podamos encontrar en el resto de la web acerca de GWT relacionado con los tres eclipse que estamos comparando, tenemos que eclipse sigue estando en la cima lo que quiere decir, que todo gira en torno a este IDE, mientras que un poco parte se relaciona con netbeans y nada con IntelliJ IDEA
(3.2.1)/3

- La licencia que pueda tener un IDE es muy importante en especial si se trata de trabajar con GWT, pues no tiene licencia por lo que es *open souce*, por tanto, eclipse y netbeans se encuentran con la misma puntuación mientras que el tercer IDE no tiene puntuación por ser licenciado. (3,3,1)/3

3.6.5.4. Calificación

$$Pc = \sum(w) = 3+3+3 = 9$$

$$PEcli = \sum(x) = 3+3+3 = 9$$

$$PNetb = \sum(y) = 2+2+3 = 7$$

$$PIntel = \sum(z) = 1+1+1 = 3$$

$$Cc - Ecli: PEcli / Pc = (9/9)*100\% = 100\%$$

$$Cc - Netb: PNetb / Pc = (7/9)*100\% = 77,78\%$$

$$Cc - Intel: PIntel / Pc = (3/9)*100\% = 33,33\%$$

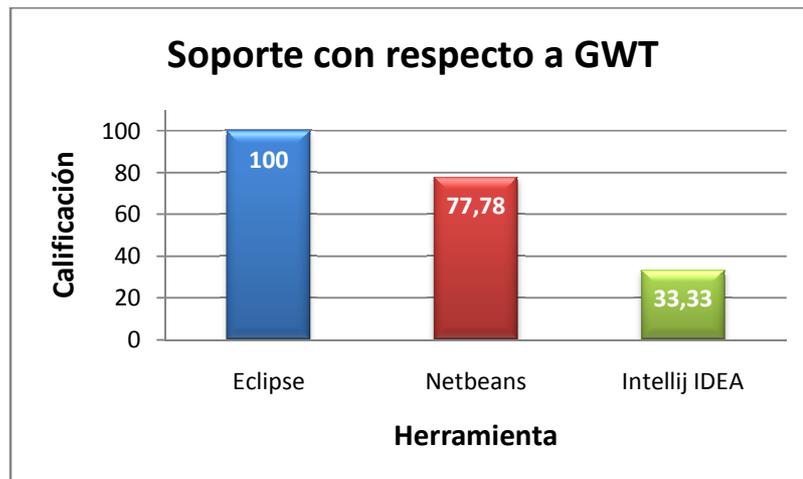


Gráfico III.13: Comparación estadística del parámetro Soporte Técnico con GWT

3.7. Puntajes alcanzados

Los puntajes finales y el porcentaje que se han obtenido una vez que se ha analizado la plataforma GWT con cada entorno de desarrollo integrado (IDE) según las variables evaluadas, se obtienen de la siguiente manera:

Puntaje Total del Análisis: (PT) = \sum (Pc)

Puntaje Total de Eclipse: (PTEcli) = \sum (PEcli)

Puntaje Total de Netbeans: (PTNetb) = \sum (PNetb)

Puntaje Total de INTELLIJ IDEA: (PTIntel) = \sum (PIntel)

Porcentaje Total de Eclipse: (% Ecli) = (PTEcli/PT) * 100%

Porcentaje Total de Netbeans: (% Netb) = (PTNetb/PT) * 100%

Porcentaje Total de INTELLIJ IDEA: (% Intel) = (PTIntel/PT) * 100%

Tabla III.9: Tabla General de Resultados

Parámetro	Variable	Eclipse	Netbeans	INTELLIJ IDEA
Acceso a la Base de Datos	Utilización de RPC.	3	3	2
	Número de Plug-in para el acceso a la base.	2	3	3
	Grado de dificultades para el acceso a la base de datos.	3	3	3
	Número de pasos que se necesitan para el acceso.	2	1	1
Compilación	Número de líneas de código.	2	3	2
	Tiempo mínimo de depuración.	3	2	2
	Capacidad para detectar errores.	3	2	2
	Capacidad para depurar errores.	3	3	2
Interfaz Gráfica de Usuario (GUI)	Diseñador visual para GUI.	1	1	1
	Generación de métodos set y get.	3	3	1

	Menor coste.	3	3	1
	Reutilización de componentes gráficos.	3	2	1
Seguridad	Encapsulamiento del código JAVA.	3	3	3
	Encapsulamiento de la página HTML.	3	3	3
	Número de Protocolos.	3	3	3
	Consistencia de las claves para el acceso al sistema.	3	3	3
Soporte Técnico de GWT	Ayuda en línea del sitio oficial de GWT.	3	2	1
	Páginas de ayuda de GWT.	3	2	1
	Licencias.	3	3	1
TOTALES		52	48	36

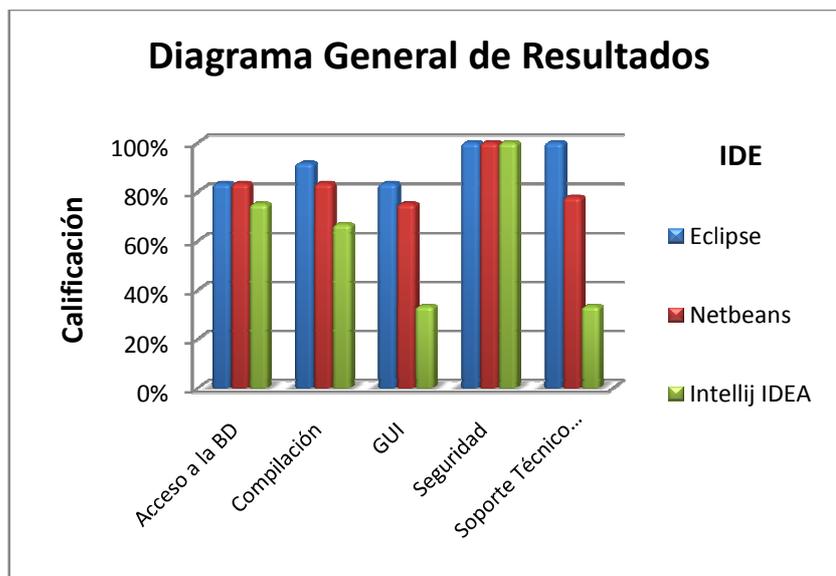


Gráfico III.14: Diagrama General de Resultados

$$PT = 12+12+12+12+9 = 57$$

$$PTEcli = 10+11+10+12+9 = 52$$

$$PTNetb = 10+10+9+12+7 = 48$$

$$PTIntel = 9+8+4+12+3 = 36$$

$$(\% \text{ Ecli}) = (52/57) * 100 = 91,22\%$$

Equivalente a Excelente

$$(\% \text{ Netb}) = (48/57) * 100 = 84,21\%$$

Equivalente a Muy Bueno

$$(\% \text{ Intel}) = (36/57) * 100 = 63,15$$

Equivalente a Regular

3.8. Interpretación

Después de haberse realizado el análisis comparativo correspondiente de los respectivos parámetros con sus variables, se ha determinado que para el uso de la plataforma GWT el IDE eclipse es el que tiene una máxima calificación del 91.22% correspondiente a excelente por lo que se determina que es el entorno más adecuado para desarrollar la aplicación web usando GWT, mientras que el IDE Netbeans obtuvo una calificación de 84.21% que es muy bueno y el INTELLIJ IDEA del 63.17% equivalente a regular.

3.9. Resultado del análisis

- Los IDE's Eclipse y el Netbeans han demostrado ser los más idóneos en los que se refiere al acceso de base de datos usando la plataforma RIA GWT, ya que según el análisis realizado son los que más puntajes han alcanzado quedando el IntelliJ IDEA en último lugar que puede ser por el motivo de ser licenciado o pagado.
- Para poder usar la plataforma RIA GWT en un IDE determinado el número de plug-in que se usa debe ser lo más mínimo posible para que la aplicación sea muy eficiente, es

por esto que eclipse mantiene la ventaja sobre los demás IDE's analizados pues aquí solo se añaden los plug-in que se requieren haciendo que la aplicación web sea más liviana y no ocupe demasiados recursos.

- En lo que se refiere a la compilación que realiza cada IDE se observa que cuando se compila una aplicación web usando la plataforma RIA GWT Eclipse resulta más conveniente que los demás IDE por tener menos tiempos de respuestas, dando al usuario una experiencia más cómoda amigable para trabajar.
- La detección y depuración de errores que se tiene usando la plataforma RIA GWT es eficiente en eclipse pues se lo puede localizar de una manera muy fácil gracias a la opción de depuración en la que se analiza paso por paso la aplicación y se la puede detener cuando el programador lo desee.
- En lo que se refiere al uso de una Interfaz Gráfica de Usuario mientras se trabaja con GWT, es difícil pues, esta plataforma no permite tal opción en ninguno de los IDE's que se tenga para el desarrollo, sin embargo, existen diseñadores gráficos que permiten trabajar con ello, con la única desventaja que tiene un costo medio elevado y solo trabajan con el IDE eclipse, por tanto en este parámetro se claramente la ventaja de este IDE sobre los demás que se tomó para el análisis.
- El coste que ofrece GWT como plataforma es gratuito pero a nivel del compilador ya tiene costo por eso es importante poder contar con un IDE que sea open source, cumpliendo con esta expectativa eclipse y netbeans.
- La seguridad que brinda GWT es muy grande y robusta pues ofrece la protección necesaria contra amenazas encapsulando tanto el código fuente de la aplicación como

el código HTML con el que se ha desarrollado garantizando al programador integridad, confidencialidad y disponibilidad.

- Cuando se habla de la consistencia que debe tener una aplicación web por lo general se refiere a que los datos ingresados desde el sistema sean los mismos que se encuentran en la base para que el inicio de sesión sea todo un éxito y esto es lo que nos proporciona GWT sin importar el IDE que se esté usando gracias a la utilidad que posee llamado callback.
- En cuanto al soporte técnico que GWT pueda poseer para ser usado en cualquier IDE es muy importante para completar el desarrollo de una aplicación web por que una vez investigado en la web se tiene que toda la información que se podría necesitar para alcanzar las metas deseadas se encuentran bajo eclipse y esto gracias a la compatibilidad que tiene con GWT.
- La licencia es otro de los factores importantes que se debe considerar para el desarrollo de una aplicación web y en particular para una empresa estatal en donde se trabajan actualmente con licencias *Open Source*, por tanto, GWT junto a eclipse brinda la posibilidad de poder hacer uso de esta ventaja permitiendo al programador un ahorro económico.
- La flexibilidad que brinda GWT para ser usado en un determinado IDE es una gran ventaja para los programadores pues permite, adaptar módulos de otros proyectos desarrollados en un IDE distinto y así lograr con esto una aplicación mucho más completa y modular.

3.10. Conclusión

Después de haber realizado el análisis comparativo y según los resultados obtenidos de todos los parámetros con sus respectivas variables, se puede concluir que para usar la plataforma RIA GWT con todas sus ventajas y desventajas es el IDE Eclipse con una puntuación de 91.22% que fue el mayor promedio con respecto a Netbeans con 84.21% y IntelliJ con 63.15%, es decir, brindando las mejores prestaciones, por tanto, es el entorno de desarrollo que se usará para el desarrollo de la aplicación web para el departamento de recursos humanos de la Refinería Estatal de Esmeraldas (REE).

3.11. Demostración de la hipótesis

Para poder realizar la demostración de la hipótesis de este trabajo investigativo se va a probar que con el uso de la tecnología RIA GWT se reducirá el tiempo en el flujo de trabajo de los procesos que realiza el departamento de Recursos Humanos de la Refinería Estatal de Esmeraldas.

Se tomará en consideración la medición de muchas variables en las que se analizará los tiempos de respuesta que se dan actualmente en el departamento de recursos humanos de la REE sin el uso de la aplicación web, así como también, los que se podrían dar usando la aplicación web para dicho departamento, las variables que se habló anteriormente se detallan a continuación aplicadas a los módulos de Consultas Generales y al módulo de las Estadísticas.

3.11.1. Tiempos de Respuesta SIN la plataforma RIA GWT

Cabe recalcar que en este parámetro a más de considerar el tiempo que se lleva en sacar los datos solicitados se debe considerar también el tiempo con el que cuenta la señora encargada de las nóminas. Además, los módulos que se va a tomar en cuenta para determinar los tiempos de respuesta sin la plataforma GWT son: el Módulo de Consultas Generales y el Módulo de Estadísticas.

3.11.1.1. Módulo de Consultas Generales

En este módulo se realiza la gestión de las acciones del personal en departamento de recursos humanos de la REE, las mismas que comprenden las siguientes consultas:

- Consultas de asistencia de los empleados.

La señora encargada de las nóminas es la encargada de realizar este trabajo, para ello tiene que realizar muchos procesos como es ir otra librería, en la cual se encuentra las tablas correspondientes a la asistencia y los permisos que se han realizado en un día determinado, y para ello se ocupa un tiempo de 10 minutos correspondientes a 600 segundos.

- Consultas de los cursos recibidos por los empleados en su vida laboral.

Para sacar los resultados de esta consulta es necesario que el empleado encargado haga una consulta para cada uno de los empleados que requieran esta información, para lo cual necesita hacer una combinación de dos tablas como son la de histórico de

capacitación y la correspondiente a los países, demorándose un tiempo de 9 minutos equivalentes a 540 segundos.

- Consultas de los datos familiares de los empleados.

En esta consulta es necesario realizar una sentencia a la tabla donde se almacena el registro de los datos de los familiares, en la que se pierde un tiempo de 5 minutos correspondientes a 300 segundos por empleado.

- Consultas de los datos personales de los empleados.

Esta es una de las consultas que más tiempo se demora por empleado, pues para sacar los datos solicitados se necesita realizar una combinación de varias tablas entre ellas tenemos: la de los datos personales del empleado, el distributivo, el del estado, entre otras, llevándose un tiempo de 14 minutos equivalentes a 840 segundos.

- Consultas del desempeño laboral en sus evaluaciones en una fecha determinada.

El proceso de esta consulta es muy exhaustivo, pues se debe primero ingresar la fecha de la evaluación en la que se contemplan el año y semestre, para luego arrojar los datos del empleado que le realizó la respectiva evaluación, así como también todos los puntajes alcanzados en los diferentes parámetros que se evalúan. En todo lo explicado anteriormente el personal autorizado consume un tiempo de 15 minutos equivalente a 900 minutos

- Consultas del historial financiero de los empleados.

Esta consulta se refiere al valor que le corresponde a cada uno de los empleados en el mes, lo cual se obtiene de la tabla correspondiente a los ingresos y descuentos ocupando para esto un tiempo de 5 minutos equivalentes a 300 segundos.

- Consultas del record académico de los empleados.

Para sacar los datos de esta consulta es necesario ingresar a dos tablas las cuales son la de la educación formal y la correspondiente a las ciudades en donde el empleado encargado utiliza un tiempo de 10 minutos equivalentes a 600 segundos por persona.

- Consultas del saldo vacacional de los empleados.

El saldo vacacional es uno de los puntos que se consideran importantes pues en él se refleja los días que le quedan de vacaciones al empleado, así como también, los días que ha ocupado. Para dar los datos solicitados se emplea un tiempo de 13 minutos por la combinación de varias tablas, siendo estos equivalentes a 780 segundos.

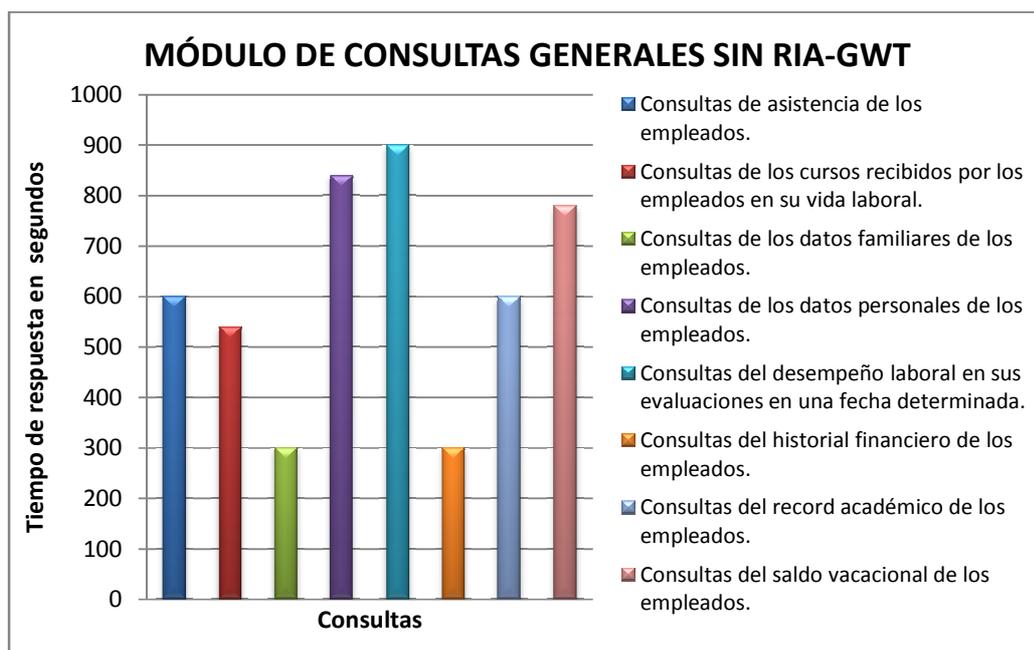


Gráfico III.15: Módulos de las Consultas Generales sin RIA-GWT

3.11.2. Módulo de Estadísticas

En la Refinería Estatal de Esmeraldas la información sobre las estadísticas se realizaba mediante el personal autorizado de Nómina, los datos se entregan en valores numéricos y no en barras estadísticas, a continuación se detallará los tiempos de respuesta de cada uno de las estadísticas que está desarrollado en el sistema operativo AS400.

Dentro de este módulo se contempla los siguientes aspectos:

- Desempeño de los empleados.

La persona autorizada de la Nómina para entregar la información sobre el desempeño de los empleados solicitaba el año y trimestre para buscar en las tablas correspondientes de la base de datos y luego ser emitir el resultado al empleado que lo solicito, el tiempo de respuesta con el sistema es de 15 minutos correspondientes a 900 segundos.

- Número de mujeres y hombres de la empresa.

El personal autorizado de la Nomina en REE al momento de realizar esta actividad se tarda un tiempo de respuesta de 5 minutos correspondientes a 300 segundos.

- Número de empleados con nivel académico.

La encargada de la Nómina en REE para obtener la información sobre los niveles académicos de los empleados debe realizar una consulta en la base de datos con los datos de primario, secundario y universitario para que el sistema arroje los valores en

cada nivel académico para entregar la información al empleado que lo solicita, el sistema demora un tiempo de respuesta de 10 minutos correspondientes a 600 segundos.

- Número de profesionales principales de la empresa.

En esta actividad la autorizada de la Nómina pregunta sobre que profesiones desea que emita la información, según los datos que solicite el empleado, la persona autorizada realiza las consultas correspondientes en la base de datos para entregar la información, el sistema tiene un tiempo de respuesta de 20 minutos correspondientes a 1200 segundos.

- Número de trabajadores contratados en el mes.

La persona autorizada de la Nómina al realizar esta actividad pregunta al empleado que lo está solicitando, trabajadores que ingresaron en una fecha determinada para realizar las consultas correspondientes, el sistema da el tiempo de respuesta de 15 minutos correspondiente a 900 segundos.

- Número de trabajadores en las diferentes áreas de trabajo en la empresa.

Al momento de realizar esta actividad la persona autorizada de la Nómina ejecuta las consultas correspondientes a la base de datos para la entrega de información, el sistema tiene un tiempo de respuesta de 25 minutos correspondientes a 1500 segundos.

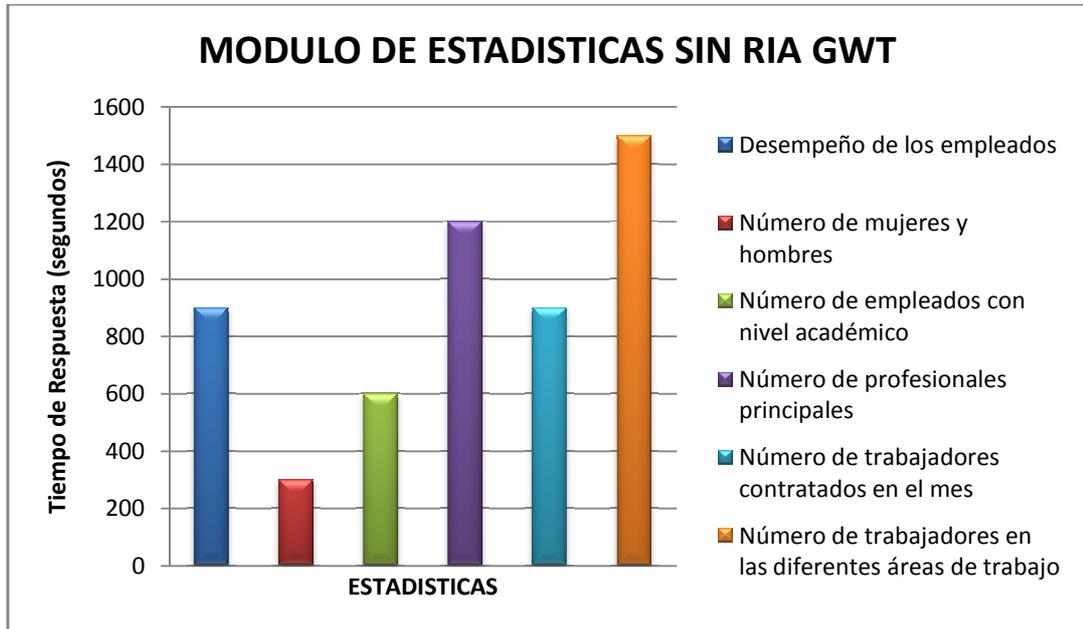


Gráfico III.16: Diagrama del Módulo de Estadísticas Sin RIA GWT

3.11.3. Tiempos de Respuestas CON la plataforma RIA GWT

Al igual que el punto anterior se va a considerar los módulos de Consultas Generales y el de Estadísticas, pero aplicado al uso de la aplicación web.

3.11.3.1. Módulo de Consultas Generales

Dentro de las consultas generales que se va a resolver con la aplicación web y que va a servir para comprobar la hipótesis planteada en esta investigación tenemos los siguientes:

- Consultas de asistencia de los empleados.

Para devolver los datos solicitados el empleado solo debe escoger esta opción que se encuentra dentro del menú consultas encontrando con ello la asistencia que ha tenido

durante el mes y los permisos correspondientes, demorándose un tiempo de 3 segundos.

- Consultas de los cursos recibidos por los empleados en su vida laboral.

En la consulta de los cursos recibidos por cada empleado, se tiene que se puede acceder de la misma forma que la anterior, ocupando un tiempo de 3 segundos aproximadamente.

- Consultas de los datos familiares de los empleados.

Esta consulta al igual que las anteriores solo se demora un tiempo de 3 segundos aproximadamente para obtener los datos solicitados.

- Consultas de los datos personales de los empleados.

Los datos personales que antes eran muy complicado resolverlo, ahora se puede notar que con la aplicación web desarrollada con la plataforma GWT resulta muy sencillo, pues solo basta con seleccionar esta opción y esperar unos 3 segundos aproximados para ver los resultados.

- Consultas del desempeño laboral en sus evaluaciones en una fecha determinada.

En esta consulta el empleado debe seleccionar la fecha de evaluación que desea consultar y luego de ello, solo basta esperar unos 3 segundos aproximadamente para obtener los datos solicitados.

- Consultas del historial financiero de los empleados.

Para obtener esta información solo basta con seleccionar esta opción en el menú de consultas, y solo tocará esperar 3 segundos aproximadamente para ver los resultados.

- Consultas del record académico de los empleados.

El record académico se lo podrá conseguir seleccionando esta opción y esperar 3 segundos aproximadamente para visualizar los resultados.

- Consultas del saldo vacacional de los empleados.

Esta consulta antes de tener esta aplicación era un poco difícil obtenerla pero ahora solo basta con seleccionar esta opción en el menú consultas y esperar unos 3 segundos aproximadamente.

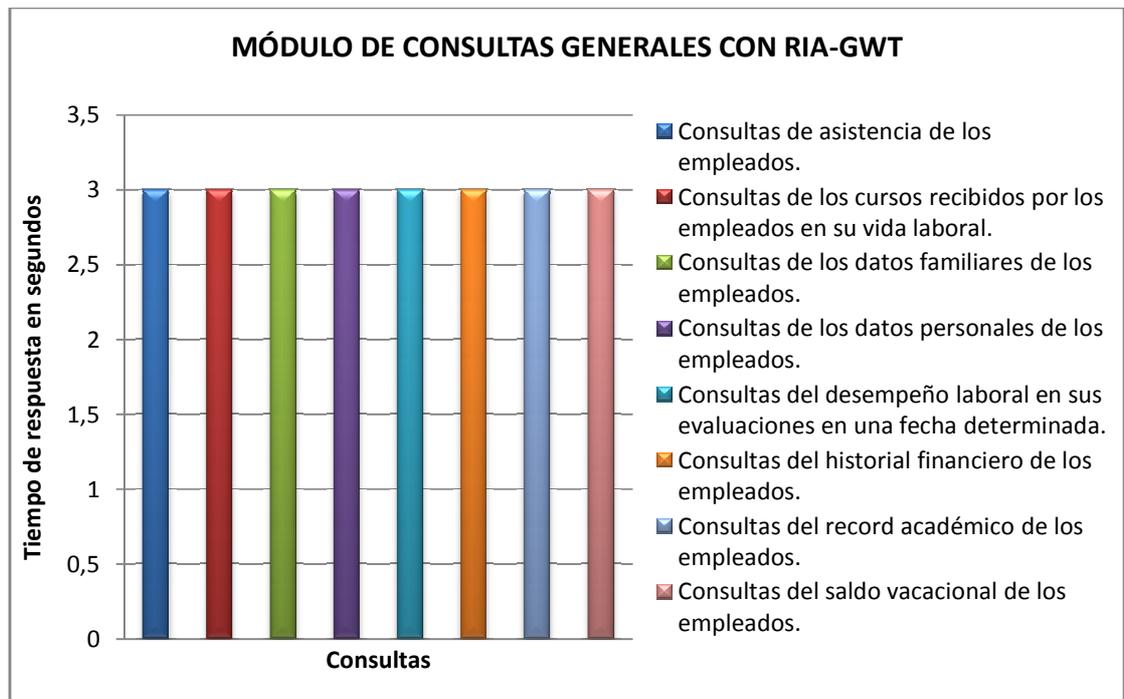


Gráfico III.17: Módulos de las Consultas Generales con RIA-GWT

3.11.3.2. Modulo de Estadísticas visualizado en barras.

En la Refinería Estatal de Esmeraldas la información sobre las estadísticas se visualizará mediante barras estadísticas y valores numéricos, que lo pueden solicitar los empleados que

están encargados de obtener dicha información con solo autenticarse al momento de ingresar a la aplicación web, a continuación se detallará los tiempos de respuesta de cada uno de las estadísticas que se mostrará en la aplicación web llamada SISREHU.

- Desempeño de los empleados.

Los empleados autorizados pueden obtener la información sobre el desempeño laboral, seleccionando el año y trimestre y se visualizará los datos de los trabajadores, la aplicación web tiene un tiempo de respuesta de tan solo 5 segundos.

- Número de mujeres y hombres de la empresa.

Los empleados autorizados al momento de seleccionar esta actividad, la aplicación se tarda un tiempo de respuesta de 3 segundos.

- Número de empleados con nivel académico.

Al momento de realizar la consulta sobre el nivel académico de los trabajadores el personal autorizado debe seleccionar dicha actividad y enseguida se mostrará la información, la misma que tiene un tiempo de respuesta de 3 segundos.

- Número de profesionales principales de la empresa.

En esta actividad se mostrará los profesionales principales, para esto se debe seleccionar la tarea a ser realizada y se obtendrá la información, en un tiempo de respuesta de 3 segundos.

- Número de trabajadores contratados en el mes.

Los empleados al momento de ejecutar dicha tarea debe seleccionar la fecha determinada en la que se desee saber el número de trabajadores que ingresaron en la

REE y además el número de trabajadores que fueron asignados a las diferentes áreas de trabajo y los datos serán visualizados, la aplicación tiene un tiempo de respuesta de 5 segundos.

- Número de trabajadores en las diferentes áreas de trabajo en la empresa.

Se obtendrá la información de esta actividad con tan solo que el empleado autorizado seleccione la opción del número de trabajadores en las diferentes áreas de trabajo, la aplicación tiene un tiempo de respuesta de 3 segundos.

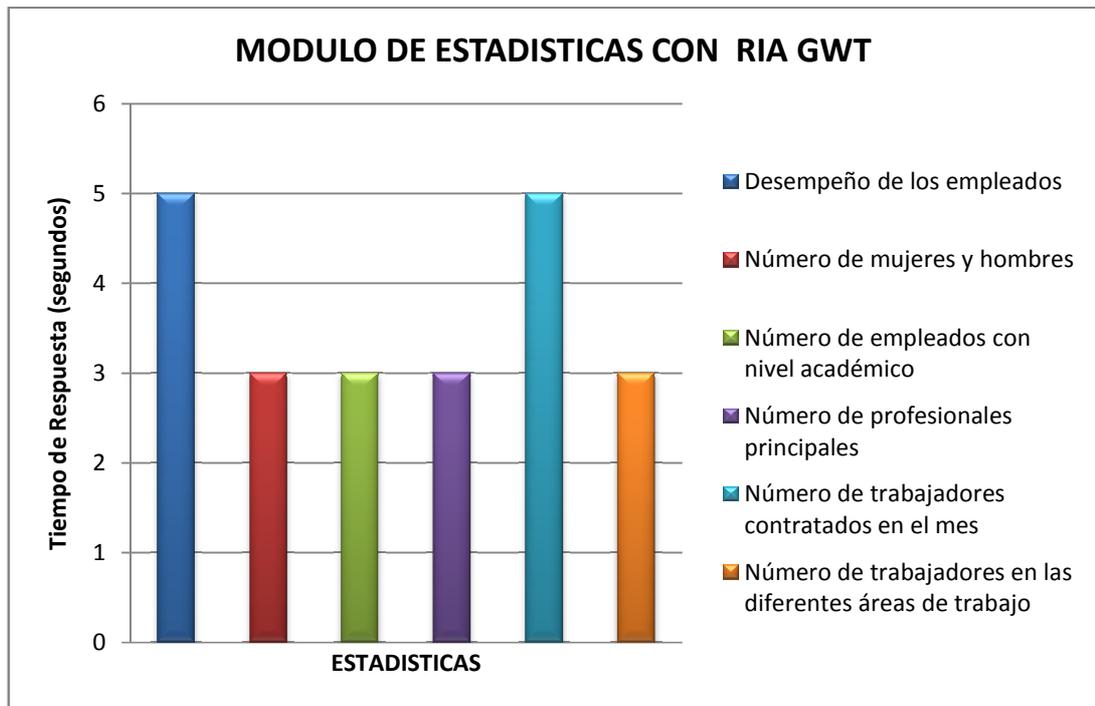


Gráfico III.18: Diagrama del Módulo de Estadísticas Con RIA GWT

3.11.4. Análisis matemático

Este análisis se lo realizará empleando fórmulas matemáticas verificando los tiempos de respuestas que se han obtenido tanto sin el uso de la plataforma RIA-GWT, así como, con

el uso de dicha plataforma para luego sacar conclusiones que nos servirán para demostrar la veracidad o no de la hipótesis, las fórmulas que se van a emplear son las siguientes:

$$\text{Puntaje Total del Análisis: (PT)} = \sum (\text{PSinGWT} + \text{PConGWT})$$

$$\text{Puntaje sin RIA-GWT} = \text{PSinGWT}$$

$$\text{Puntaje con RIA-GWT} = \text{PConGWT}$$

$$\text{Tiempo ahorrado} = \text{PSinGWT} - \text{PConGWT}$$

3.11.4.1. Módulo de Consultas Generales

Los aspectos que aquí se va a considerar tanto sin el uso de RIA-GWT como con el uso de RIA-GWT son los siguientes:

- Consultas de asistencia de los empleados.

$$\text{PT} = \sum (\text{PSinGWT} + \text{PConGWT})$$

$$\text{PT} = 640 + 3$$

$$\text{PT} = 643$$

$$\text{PSinGWT} = (\text{PSinGWT} * 100 \%) / \text{PT}$$

$$\text{PSinGWT} = (640 * 100\%) / 643$$

$$\text{PSinGWT} = 99.53 \%$$

$$\text{PConGWT} = (\text{PConGWT} * 100 \%) / \text{PT}$$

$$\text{PConGWT} = (3 * 100\%) / 643$$

$$\text{PConGWT} = 0.47 \%$$

$$\text{Tiempo ahorrado} = \text{PSinGWT} - \text{PConGWT}$$

$$\text{Tiempo ahorrado} = 99.53\% - 0.47 \%$$

Tiempo ahorrado = 99.06 %

- Consultas de los cursos recibidos por los empleados en su vida laboral.

$$PT = \sum (PSinGWT + PConGWT)$$

$$PT = 540 + 3$$

$$PT = 543$$

$$PSinGWT = (PSinGWT * 100 \%) / PT$$

$$PSinGWT = (540 * 100\%) / 543$$

$$PSinGWT = 99.45 \%$$

$$PConGWT = (PConGWT * 100 \%) / PT$$

$$PConGWT = (3 * 100\%) / 543$$

$$PConGWT = 0.55 \%$$

$$\text{Tiempo ahorrado} = PSinGWT - PConGWT$$

$$\text{Tiempo ahorrado} = 99.45\% - 0.55 \%$$

$$\text{Tiempo ahorrado} = 98.9 \%$$

- Consultas de los datos familiares de los empleados.

$$PT = \sum (PSinGWT + PConGWT)$$

$$PT = 300 + 3$$

$$PT = 303$$

$$PSinGWT = (PSinGWT * 100 \%) / PT$$

$$PSinGWT = (300 * 100\%) / 303$$

$$PSinGWT = 99.01 \%$$

$$\mathbf{PConGWT = (PConGWT * 100 \%) / PT}$$

$$PConGWT = (3 * 100\%) / 303$$

$$PConGWT = 0.99 \%$$

$$\mathbf{Tiempo ahorrado = PSinGWT - PConGWT}$$

$$Tiempo ahorrado = 99.01\% - 0.99 \%$$

$$Tiempo ahorrado = 98.02\%$$

- Consultas de los datos personales de los empleados.

$$\mathbf{PT = \sum (PSinGWT + PConGWT)}$$

$$PT = 840 + 3$$

$$PT = 843$$

$$\mathbf{PSinGWT = (PSinGWT * 100 \%) / PT}$$

$$PSinGWT = (840 * 100\%) / 843$$

$$PSinGWT = 99.64 \%$$

$$\mathbf{PConGWT = (PConGWT * 100 \%) / PT}$$

$$PConGWT = (3 * 100\%) / 843$$

$$PConGWT = 0.36 \%$$

$$\mathbf{Tiempo ahorrado = PSinGWT - PConGWT}$$

$$Tiempo ahorrado = 99.64\% - 0.36 \%$$

$$Tiempo ahorrado = 99.28 \%$$

- Consultas del desempeño laboral en sus evaluaciones en una fecha determinada.

$$\mathbf{PT = \sum (PSinGWT + PConGWT)}$$

$$PT = 900 + 3$$

$$PT = 903$$

$$\mathbf{PSinGWT = (PSinGWT * 100 \%) / PT}$$

$$PSinGWT = (900 * 100\%) / 903$$

$$PSinGWT = 99.66 \%$$

$$\mathbf{PConGWT = (PConGWT * 100 \%) / PT}$$

$$PConGWT = (3 * 100\%) / 903$$

$$PConGWT = 0.34 \%$$

$$\mathbf{T tiempo ahorrado = PSinGWT - PConGWT}$$

$$\text{T tiempo ahorrado} = 99.66\% - 0.34 \%$$

$$\text{T tiempo ahorrado} = 99.32 \%$$

- Consultas del historial financiero de los empleados.

$$\mathbf{PT = \Sigma (PSinGWT + PConGWT)}$$

$$PT = 300 + 3$$

$$PT = 303$$

$$\mathbf{PSinGWT = (PSinGWT * 100 \%) / PT}$$

$$PSinGWT = (300 * 100\%) / 303$$

$$PSinGWT = 99.01 \%$$

$$\mathbf{PConGWT = (PConGWT * 100 \%) / PT}$$

$$PConGWT = (3 * 100\%) / 543$$

$$PConGWT = 0.99 \%$$

$$\mathbf{T tiempo ahorrado = PSinGWT - PConGWT}$$

$$\text{T tiempo ahorrado} = 99.01 \% - 0.99 \%$$

Tiempo ahorrado = 98.02 %

- Consultas del record académico de los empleados.

$$PT = \sum (PSinGWT + PConGWT)$$

$$PT = 600 + 3$$

$$PT = 603$$

$$PSinGWT = (PSinGWT * 100 \%) / PT$$

$$PSinGWT = (600 * 100\%) / 603$$

$$PSinGWT = 99.5 \%$$

$$PConGWT = (PConGWT * 100 \%) / PT$$

$$PConGWT = (3 * 100\%) / 603$$

$$PConGWT = 0.5 \%$$

$$\text{Tiempo ahorrado} = PSinGWT - PConGWT$$

$$\text{Tiempo ahorrado} = 99.5\% - 0.5 \%$$

$$\text{Tiempo ahorrado} = 99 \%$$

- Consultas del saldo vacacional de los empleados.

$$PT = \sum (PSinGWT + PConGWT)$$

$$PT = 780 + 3$$

$$PT = 783$$

$$PSinGWT = (PSinGWT * 100 \%) / PT$$

$$PSinGWT = (780 * 100\%) / 783$$

$$PSinGWT = 99.61 \%$$

$$\mathbf{PConGWT = (PConGWT * 100 \%) / PT}$$

$$PConGWT = (3 * 100\%) / 783$$

$$PConGWT = 0.39 \%$$

$$\mathbf{Tiempo ahorrado = PSinGWT - PConGWT}$$

$$Tiempo ahorrado = 99.61\% - 0.39 \%$$

$$Tiempo ahorrado = 99.22 \%$$

3.11.4.2. Módulo de Estadísticas

En esta sección se realizará los cálculos respectivos de cada actividad.

- Desempeño de los empleados.

$$\mathbf{PT = \sum (PSinGWT + PConGWT)}$$

$$PT = 900 + 5$$

$$PT = 905$$

$$\mathbf{PSinGWT = (PSinGWT * 100 \%) / PT}$$

$$PSinGWT = (900 * 100\%) / 905$$

$$PSinGWT = 94.45 \%$$

$$\mathbf{PConGWT = (PConGWT * 100 \%) / PT}$$

$$PConGWT = (5 * 100\%) / 905$$

$$PConGWT = 0.55 \%$$

$$\mathbf{Tiempo ahorrado = PSinGWT - PConGWT}$$

$$Tiempo ahorrado = 94.45\% - 0.55 \%$$

$$Tiempo ahorrado = 98.90\%$$

- Número de mujeres y hombres de la empresa

$$PT = \Sigma (PSinGWT + PConGWT)$$

$$PT = 300 + 3$$

$$PT = 303$$

$$PSinGWT = (PSinGWT * 100 \%) / PT$$

$$PSinGWT = (300 * 100\%) / 303$$

$$PSinGWT = 99 \%$$

$$PConGWT = (PConGWT * 100 \%) / PT$$

$$PConGWT = (3 * 100\%) / 303$$

$$PConGWT = 1 \%$$

$$\text{Tiempo ahorrado} = PSinGWT - PConGWT$$

$$\text{Tiempo ahorrado} = 99\% - 1 \%$$

$$\text{Tiempo ahorrado} = 98\%$$

- Número de empleados con nivel académico

$$PT = \Sigma (PSinGWT + PConGWT)$$

$$PT = 600 + 3$$

$$PT = 603$$

$$PSinGWT = (PSinGWT * 100 \%) / PT$$

$$PSinGWT = (600 * 100\%) / 603$$

$$PSinGWT = 99.5\%$$

$$PConGWT = (PConGWT * 100 \%) / PT$$

$$PConGWT = (3 * 100\%) / 603$$

$$PConGWT = 0.5 \%$$

$$\text{Tiempo ahorrado} = \text{PSinGWT} - \text{PConGWT}$$

$$\text{Tiempo ahorrado} = 99.5\% - 0.5\%$$

$$\text{Tiempo ahorrado} = 99\%$$

- Número de profesionales principales de la empresa

$$\text{PT} = \Sigma (\text{PSinGWT} + \text{PConGWT})$$

$$\text{PT} = 1200 + 3$$

$$\text{PT} = 1203$$

$$\text{PSinGWT} = (\text{PSinGWT} * 100\%) / \text{PT}$$

$$\text{PSinGWT} = (1200 * 100\%) / 1203$$

$$\text{PSinGWT} = 99.75\%$$

$$\text{PConGWT} = (\text{PConGWT} * 100\%) / \text{PT}$$

$$\text{PConGWT} = (3 * 100\%) / 1203$$

$$\text{PConGWT} = 0.25\%$$

$$\text{Tiempo ahorrado} = \text{PSinGWT} - \text{PConGWT}$$

$$\text{Tiempo ahorrado} = 99.75\% - 0.25\%$$

$$\text{Tiempo ahorrado} = 99.5\%$$

- Número de trabajadores contratados en el mes

$$\text{PT} = \Sigma (\text{PSinGWT} + \text{PConGWT})$$

$$\text{PT} = 900 + 5$$

$$\text{PT} = 905$$

$$\mathbf{PSinGWT = (PSinGWT * 100 \%) / PT}$$

$$PSinGWT = (900 * 100\%) / 905$$

$$PSinGWT = 94.45 \%$$

$$\mathbf{PConGWT = (PConGWT * 100 \%) / PT}$$

$$PConGWT = (5 * 100\%) / 905$$

$$PConGWT = 0.55 \%$$

$$\mathbf{Tiempo ahorrado = PSinGWT - PConGWT}$$

$$Tiempo ahorrado = 94.45\% - 0.55 \%$$

$$Tiempo ahorrado = 98.90\%$$

- Número de trabajadores en las diferentes áreas de trabajo en la empresa

$$\mathbf{PT = \Sigma (PSinGWT + PConGWT)}$$

$$PT = 1500 + 3$$

$$PT = 1503$$

$$\mathbf{PSinGWT = (PSinGWT * 100 \%) / PT}$$

$$PSinGWT = (1500 * 100\%) / 1503$$

$$PSinGWT = 99.8 \%$$

$$\mathbf{PConGWT = (PConGWT * 100 \%) / PT}$$

$$PConGWT = (3 * 100\%) / 1503$$

$$PConGWT = 0.2 \%$$

$$\mathbf{Tiempo ahorrado = PSinGWT - PConGWT}$$

$$Tiempo ahorrado = 99.8\% - 0.2 \%$$

$$Tiempo ahorrado = 99.60\%$$

3.11.5. Interpretación de los Resultados

A continuación se interpretará los resultados obtenidos en el módulo de las consultas generales:

- Se puede observar que para la consulta de asistencia de los empleados usando la plataforma RIA-GWT se ha reducido en un 98.9% del tiempo empleado por el personal autorizado para realizar esta gestión.
- Para realizar las consultas de los cursos recibidos por parte de los empleados, se ha visto que usando RIA-GWT se pudo ahorrar 98.9% del tiempo del empleado encargado de las nóminas.
- En lo referente a las consultas de los datos de los familiares se puede ver claramente que el ahorro de tiempo con el uso de la plataforma RIA-GWT ha sido de un 98.02%.
- En cuanto a la consulta de los datos personales de los empleados se ha logrado ahorrar un tiempo de 99,28%.
- El desempeño laboral usando GWT ha resultado mucho más fácil sabiendo que el personal autorizado para esta función ha podido ahorrar 99.32% de su tiempo.
- Para la consulta del historial financiero también se puede concluir que se ha podido reducir el tiempo en este proceso en un 98.02% aproximadamente del tiempo de trabajo del personal encargado.
- El récord académico igual que los casos anteriores se ha podido reducir su tiempo, en un 99%.

- Finalmente, se puede observar cómo se reduce el tiempo para la consulta del saldo vacacional el mismo que con el uso de la aplicación web se consigue un ahorro del 99.22%.

A continuación se interpretará los resultados obtenidos en el módulo de estadísticas:

- Con respecto al desempeño de los empleados los resultados de los tiempos Sin RIA GWT es de 900 segundos es decir un 99.45% ya que el personal autorizado de la Nómina se encarga de entregar la información a los empleados con respecto a la REE y además realizar su trabajo es decir que esto le implica tiempo; y Con RIA GWT es de 5 segundos equivalente a 0.55% ya que cada empleado autorizado a la información de la REE consulta directamente en la aplicación SISREHU, por lo tanto se reduce el tiempo de respuesta un 98.90%, ayudando al empleado a optimizar su tiempo, cumpliendo así con la hipótesis.
- En el número de mujeres y hombres de la empresa se observa los tiempos de respuesta Sin RIA GWT que es de 300 segundos equivalente al 99% y Con RIA GWT de 3 segundos correspondiente al 1%, es decir, se demora más tiempo con el sistema AS400 ya que maneja solo el personal de Nómina encargada de entregar la información correspondiente a la REE a los empleados y además tiene sus propias tareas que realizar por este motivo se tarda en entregar la información, por lo tanto, se reduce el tiempo en un 98%.

- El resultado de la actividad del número de empleados con nivel académico arroja los siguientes datos, Sin RIA GWT un tiempo de 600 segundos correspondiente al 99.50%, utilizando un tiempo mayor al entregar la información ya que no todos los empleados tienen acceso a la información de la base de datos de REE, y Con RIA GWT un tiempo de 3 segundos equivalente al 0.50% ya que el empleado que tenga asignado los permisos podrá acceder que son las personas que necesitan la información y se obtendrá mediante la aplicación, reduciendo el tiempo de respuesta del 99%, cumpliendo con la hipótesis.
- Los datos que arrojo como resultado de el número de profesionales principales de la empresa es Sin RIA GWT de 1200 segundos equivalente a un 99.75% y Con RIA GWT de 3 segundos correspondiente a 0.25%, es decir reduciendo el tiempo en el 99.50%, ya que al usar la aplicación SISREHU ayuda al empleado a obtener la información en instantes mientras que con el sistema AS400 tarda en recibir la información porque lo realiza mediante el personal de la Nómina.
- Al momento de obtener la información sobre el número de trabajadores contratados en el mes se analizaron los datos Sin RIA GWT y Con RIA GWT y dio como resultado un tiempo de 900 segundos equivalente a 99.45% y 5 segundos a un 0.55% respectivamente, por lo tanto, reduciendo el tiempo en el 98.90%, es decir, el Sin RIA GWT necesita más tiempo ya que se utiliza el sistema AS400 que es manejado por el personal de Nómina, el mismo que no tiene solo esa tarea sino su propio trabajo, es decir, tarda en entregar la información.

- Los resultados obtenidos en la actividad de el número de trabajadores en las diferentes áreas de trabajo en la empresa arrojó los siguientes datos, Sin RIA GWT un tiempo de 1500 segundos equivalente al 99.80%, el mismo que se demora el personal de la Nómina en entregar la información al empleado solicitante, ya que tiene que realizar las consultas correspondientes en el sistema AS400 para obtener los datos y Con RIA GWT tarda un tiempo de 3 segundos correspondiente a un 0.20%, por lo tanto el tiempo se reduce un 99.60%, cumpliendo así con la hipótesis.

3.11.6. Conclusión

Luego de haber realizado todos los cálculos correspondientes para la demostración de la hipótesis analizando proceso por proceso se puede concluir que efectivamente con el desarrollo de la aplicación web usando la plataforma RIA-GWT para el departamento de recursos humanos de la Refinería Estatal de Esmeraldas se ha podido reducir el tiempo en el flujo de trabajo en lo concerniente a los procesos que realiza día a día los empleados de dicha entidad, permitiendo al encargado de las nóminas poder realizar su trabajo que en la actualidad es imposible por las consultas constantes que debe atender.

CAPÍTULO IV METODOLOGÍA DE ANÁLISIS

4.1. Introducción

Una propuesta metodológica para el desarrollo de una aplicación web usando GWT es de suma importancia para un programador que va a empezar a trabajar con dicha plataforma, pues con ella se va a guiar para establecer luego las fases del proyecto. Para alcanzar todo esto y en vista al objetivo propuesto en el marco referencial es necesario utilizar el método FODA.

Conocido originalmente con el método “OPEDEPO PF”, es un acróstico para identificar las Debilidades, Oportunidades, Peligros y Potencialidades, fundamentales para visualizar panoramas de cualquier ámbito, cobertura y situación, aplicable a Empresas, Instituciones o cualquier tipo de organización.

Las siglas originales eran demasiado complejas por lo que con posterioridad resultó más fácil y manejable el término FODA, convirtiéndose prácticamente en algo indispensable para analizar cualquier situación.

En el presente capítulo se elaborará mediante la utilización de una herramienta de planeación estratégica llamada FODA (Fortalezas, Oportunidades, Debilidades, Amenazas), una propuesta metodología con el fin de determinar las ventajas y desventajas que los desarrolladores de aplicaciones web pueden tener al usar la plataforma RIA GWT.

4.2. Concepto FODA

Para este caso el método FODA se orienta principalmente al análisis y resolución de problemas y se lleva a cabo para identificar y analizar las Fortalezas y Debilidades de la plataforma de desarrollo RIA GWT (consolidando las fortalezas y minimizando las debilidades), así como también, las Oportunidades (aprovechadas y no aprovechadas) y Amenazas reveladas por la información obtenida del contexto externo que pueda tener (aprovechando las ventajas de las oportunidades y eliminando o reduciendo las amenazas).

El análisis FODA permite llevar a cabo una serie de estrategias de acuerdo con la importancia de cada uno de los factores internos y externos, así como ordenarlas de acuerdo con la relación que exista entre cada una de ellas.

4.3. Componentes del FODA

Para determinar los componentes que tiene un FODA, se va a considerar el siguiente gráfico en donde se explica claramente:

	Positivos	Negativos
Internos	Fortalezas	Debilidades
Externos	Oportunidades	Amenazas

Grafico IV.1: Componentes del FODA

Las Fortalezas y Debilidades se refieren a todo lo interno que tiene la plataforma de desarrollo GWT, mientras que las Oportunidades y Amenazas son factores externos sobre los cuales la plataforma GWT no tiene control alguno. Por tanto, deben analizarse las condiciones del FODA en el siguiente orden:

- a. Fortalezas.
- b. Oportunidades.
- c. Amenazas.
- d. Debilidades.

Al detectar primero las amenazas que las debilidades, la plataforma tendrá que poner atención a las primeras y desarrollar las estrategias convenientes para contrarrestarlas, y con ello, ir disminuyendo el impacto de las debilidades. Al tener conciencia de las amenazas, la plataforma aprovechará de una manera más integral tanto sus fortalezas como sus oportunidades.

El FODA como técnica de planeación, permitirá contar con información valiosa proveniente de otras plataformas que se dedican a la misma actividad. Además requiere del análisis de los diferentes elementos que forman parte del funcionamiento interno de GWT y que puedan tener implicaciones en su desarrollo, como pueden ser las ventajas y desventajas que esta plataforma presenta ante otras.

4.4. Diagnóstico de luz y Fuerza del Centro (LFC)

En este diagnóstico se evalúa los dos tipos de análisis que se puede tener en el método FODA para la metodología que se va a plantear, estos análisis son internos y externos, los cuales se los explica de una mejor manera en el siguiente gráfico:

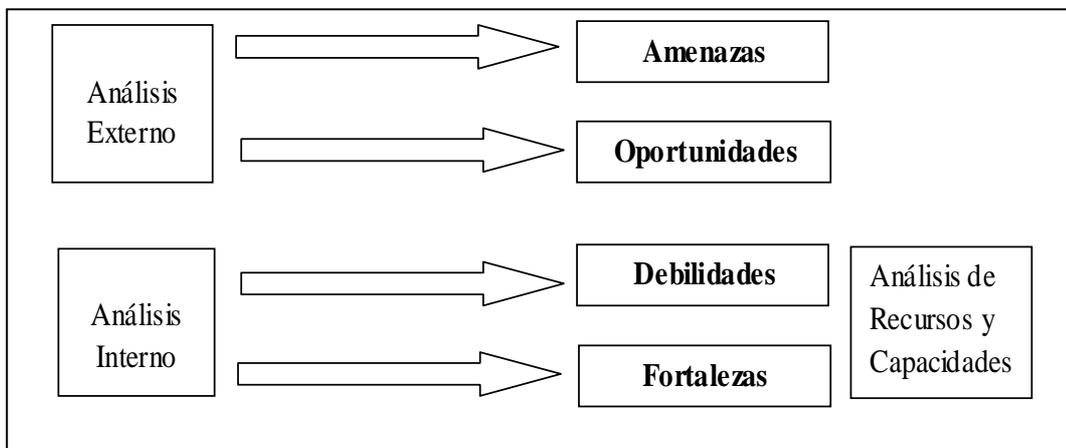


Gráfico IV.2: Tipos de análisis del FODA

4.5. Propuesta metodológica

Para cumplir con uno de los objetivos del presente trabajo investigativo el cual trata de proponer una metodología que permita determinar las ventajas y desventajas de la

plataforma GWT, los autores han considerado utilizar la metodología FODA aunque es usada generalmente en las empresas para prevenir los factores que hacen que la misma decaiga y sacarla adelante, de ahí, el trabajo de los autores en adaptarlo al cumplimiento del objetivo. Se ha tomado como base una metodología usada en la siguiente página <http://www.scribd.com/doc/32395653/LFC35-Anex-01-Conceptos-de-Metodologia-FODA>, la cual propone muchas fases, las mismas que han sido resumidas de acuerdo a nuestro propósito en las etapas que a continuación se describen.

4.6. Desarrollo de las fases de la metodología

En este apartado se describirá una propuesta metodología para llevar a cabo el FODA para GWT, la cual contempla las siguientes etapas como se muestra en el siguiente gráfico:

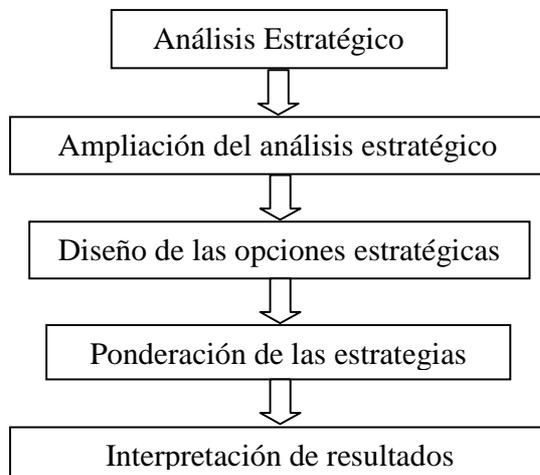


Gráfico IV.3: Etapas del proceso de análisis FODA.

4.6.1. Breve descripción de las fases

4.6.1.1. Análisis Estratégico

En esta fase se va a realizar un análisis de los factores internos y externos que comprende la plataforma RIA-GWT, en los cuales se analizará las fortalezas, oportunidades, debilidades y amenazas de la misma.

4.6.1.2. Ampliación del análisis estratégico

Aquí se medirá cada uno de los criterios de valoración mediante una matriz de confrontación, la cual va a estar representada por la relación entre las amenazas, oportunidades, fortalezas y debilidades, evaluando en ella las ventajas y desventajas de la plataforma GWT combinando sus factores internos y externos.

Al final se elaborará una matriz modificada en la que solo se tomará en cuenta los puntos en los que más interés hay que poner para que la plataforma GWT surja y sea compacta mediante la elaboración de estrategias.

4.6.1.3. Ponderación de las estrategias

En este punto se va a determinar la importancia relativa que van a tener las estrategias consideradas en el punto anterior de acuerdo a varios aspectos como la determinación del peso y la valoración de las estrategias de cada grupo.

4.6.1.4. Interpretación de los resultados

Finalmente se dará un criterio sobre todos los resultados obtenidos en las tablas de la ponderación de las estrategias, haciendo un análisis exhaustivo y llegando a la mejor conclusión para saber si con el análisis FODA se logró o no mejorar la plataforma de desarrollo RIA-GWT.

4.6.2. Análisis Estratégico

4.6.2.1. Análisis Interno

4.6.2.1.1. Fortalezas

Es una posición favorable que posee la plataforma RIA GWT, es decir, van a ser todas las ventajas que ésta posea en relación con las demás plataformas que realizan la misma función, las cuales, van a servir para responder eficazmente ante una oportunidad o ante una amenaza que pueda tener en el transcurso de la aplicación web. Entre las principales fortalezas que cuenta la plataforma de desarrollo RIA GWT y que son de mucha ayuda para el programador tenemos las siguientes:

- **F1: No requiere instalación.-** la aplicación es descargada y ejecutada automáticamente en el navegador. El software que corre actualmente la aplicación es instalada en la máquina del cliente.
- **F2: Las actualizaciones son automáticas.-** las versiones nuevas de la plataforma también se descargarán automáticamente con sólo volver a visitar la página de la aplicación web.

- **F3: Plataforma independiente.-** una aplicación rica de Internet puede ejecutarse y operar potencialmente en cada plataforma de un navegador.
- **F4: Más seguridad.-** las aplicaciones se ejecutan en un ambiente restringido en el navegador y por lo tanto son menos probables que sea dañada como las aplicaciones que necesitan ser instalados.
- **F5: Más respuesta.-** Porque no todas las acciones de los usuarios requieren comunicación con el servidor, las RIA tienden a ser más sensibles que las aplicaciones web clásicas.
- **F6: Más escalable.-** Una gran parte del trabajo computacional, así como también el cuidado estatal puede ser descargado desde el cliente, por lo que el servidor puede manejar muchos más usuarios.
- **F7: Más eficiencia de red.-** En las aplicaciones web clásicas, cada acción del usuario requiere que el servidor cargue toda la página y enviarla a la red. En el caso de las aplicaciones ricas en Internet (RIA) GWT, toda la aplicación de interfaz de usuario sólo tiene que ser comunicado una vez y todas las otras peticiones al servidor requieren sólo los datos actuales que se envían al cliente.
- **F8: Componentes de la interfaz de usuario dinámicos y reutilizables.-** Un sistema de UI widgets estándar que parece bueno, flexible, que se contempla para trabajar en todos los navegadores importantes (incluyendo safari y opera).
- **F9: RPC realmente fácil.-** Para comunicarse desde el navegador que lanza la aplicación con el servidor web, solamente se necesita definir clases de Java serializables para las peticiones y respuestas. En producción, GWT serializa

automáticamente las peticiones del navegador y de-serializa las respuestas desde el servidor web. El mecanismo de RPC de GWT puede incluso manejar jerarquía de polimorfismo en clases, y puede manejar las posibles excepciones.

- **F10: Administración del historial del navegador.-** Las aplicaciones en AJAX no necesitan utilizar el botón “Atrás” (o Back) del navegador. Y GWT no es la excepción, es decir, no es necesario que se llame a otras páginas para realizar las diferentes acciones, ni recargar el navegador.
- **F11: Depuración en tiempo real.-** Para cuando la aplicación esté lista, el código de la misma es traducido a JavaScript, pero mientras lo estás desarrollando este corre sobre una máquina virtual de Java (JVM). Lo que significa que en la fase de desarrollo se tiene la posibilidad de depurar la aplicación con los avanzados sistemas de debugging y manipulación de excepciones incluidos en IDEs como Eclipse.
- **F12: Compatibilidad con los navegadores.-** Las aplicaciones en GWT serán automáticamente soportadas por navegadores como FireFox, Internet Explorer, Mozilla, Safari, y Opera sin ningún tipo de operación para la detección de los mismos, en la mayoría de los casos.
- **F13: Integración con JUnit.-** Mediante la integración de Junit en GWT, se puede probar las aplicaciones y depurarlas en un navegador mientras se construyen. Se puede testear llamadas asíncronas a procedimientos remotos RPC.
- **F14: Internacionalización.-** Se crea aplicaciones y librerías de Internacionalización rápida y fácilmente.

- **F15: Interoperabilidad y control.-** Si las librerías de clases de GWT no son suficientes para lo que se necesita, se puede mezclar JavaScript en el código de la aplicación usando la interfaz nativa de scripts de Java (JavaScript Native Interface, JSNI).
- **F16: GWT es un proyecto de código abierto.-** Todo el código de GWT está disponible bajo la licencia Apache 2.0.

4.6.2.1.2. Debilidad

Es una posición desfavorable que tiene la plataforma RIA GWT con respecto a alguno de sus elementos y que la ubica en condiciones de no poder responder eficazmente a las oportunidades y amenazas del entorno. Entre las principales debilidades o desventaja que tiene esta plataforma tenemos:

- **D1: Requiere JavaScript o un plug-in específico.-** Porque toda la aplicación se ejecuta en el cliente a través del intérprete de JavaScript. Cuando el usuario desactiva JavaScript por completo la aplicación por lo general hace poco o nada. Obviamente es posible tener un plan alternativo para esos usuarios, pero luego se va a mantener dos aplicaciones separadas, que está lejos de ser ideal.
- **D2: No hay acceso a los recursos del sistema.-** Como las aplicaciones AJAX se ejecutan dentro de un navegador, ellos están limitados en los recursos que pueden acceder. Por ejemplo, una aplicación AJAX no puede tener acceso el sistema de archivos del cliente.

- **D3: Es duro de indexar completamente para los motores de búsqueda.**- Porque la mayoría de los motores de búsqueda no soportan que aplicaciones actualicen parcialmente las páginas o usen un plug-in específico tales como Flash. La mayoría de las aplicaciones dinámicas de Internet están mal indexadas por los motores de búsqueda.
- **D4: Cuestiones de accesibilidad.**- Al hacer actualizaciones parciales de página mediante JavaScript o un plug-in específico puede romper la accesibilidad. El problema más grande y más notorio es que no se pueden manejar correctamente la pantalla.
- **D5: Depende de una conexión a Internet.**- Porque estas aplicaciones son servidas desde la web y ejecutadas en el navegador web, por lo que requieren por lo menos una conexión a Internet inicial. Pero incluso durante su uso, es necesaria una conexión a Internet para comunicarse con el servidor. Cuando la conexión no está temporalmente disponible, las RIA's GWT no para funcionar.
- **D6:** Solo pueden ser usadas por desarrolladores JAVA. La curva de aprendizaje es lenta al principio si el programador tiene mucha experiencia en otra tecnología.
- **D7:** Dependiendo de las características de la aplicación puede ser un poco más costosa el despliegue.
- **D8: Compilación.**- Demora demasiado tiempo al momento que se compila desde el navegador propio de GWT a cualquiera de los browser que se desee presentar o ejecutar.

4.6.2.2. Análisis Externo

4.6.2.2.1. Amenazas

Se refiere a la situación desfavorable, actual o futura, que presenta el entorno de la plataforma RIA GWT, la cual debe ser enfrentada para evitar o minimizar los daños potenciales sobre el desempeño de la misma.

Entre las principales amenazas que enfrenta GWT con respecto a las demás plataformas de desarrollo tenemos las siguientes:

- **A1:** No es el framework más popular de la comunidad java por lo que existen infinidad de material disponible en la web.
- **A2:** Documentación de otros framework muy buena.
- **A3:** Crear sitios internacionales de manera rápida y efectiva.
- **A4:** Curva de aprendizaje muy baja con relación a GWT.
- **A5:** Desarrollo de componentes propios en otros framework.
- **A6:** Soporte oficial.
- **A7:** Desarrollo con herramientas RAD.
- **A8:** Separación clara del contenido de la presentación y de la lógica.
- **A9:** No es necesario conocer el framework en detalle para poder comenzar a utilizarlo.

4.6.2.2.2. Oportunidades

Es una situación favorable, actual o futura, que ofrece el entorno para la plataforma RIA GWT, cuyo aprovechamiento adecuado u oportunidad mejoraría su posición de competencia.

Entre las principales oportunidades que nos presenta el mercado para esta plataforma que es objeto de estudio tenemos:

- **O1:** Existen mayores posibilidades de trabajo, por el simple hecho de que en la actualidad no existen muchas personas que sepan el manejo del framework GWT, habiendo aproximadamente 4 personas en el Ecuador que entienden sobre esto. Por lo tanto las fuentes de trabajo que se abren son muchas en el mercado tanto nacional como internacional.
- **O2:** Posibilidad de realizar aplicaciones en empresas públicas, sabiendo que en la actualidad lo que más buscan son aplicaciones web que sean *open source*.
- **O3:** Posibilidad de realizar aplicaciones web en empresas privadas de gran envergadura, y que cuentan con las más grandes tecnologías.
- **O4:** Posibilidad de realizar las aplicaciones más rápidas en relación con el uso de otras plataformas que se dediquen a la misma actividad.
- **O5:** Para una persona o empresa que está empezando, es de muy buena alternativa empezar con esta herramienta por el simple hecho de ser gratis.

4.6.3. Ampliación del análisis estratégico

4.6.3.1. Criterios de valoración de la matriz de confrontación

Tabla IV.1: Criterios de valoración de la matriz de confrontación

	Amenazas	Oportunidades
Fortalezas	1. (--) muy negativa disminuye significativamente la fortaleza 2. (-) negativa disminuye la fortaleza 3. (0) sin relación	1. (++) Muy positiva se potencia significativamente la fortaleza 2. (+) positiva se potencia la fortaleza 3. (0) sin relación
Debilidades	1. (--) muy negativa potencia significativamente la debilidad 2. (-) negativa disminuye la debilidad 3. (0) nula sin relación	1. (++) Muy positiva disminuye significativamente la debilidad 2. (+) positiva disminuye la debilidad 3. (0) sin relación

4.6.3.2. Elaboración de la matriz de confrontación

Es una herramienta para combinar los factores internos con los factores externos de las ventajas y desventajas de la plataforma de desarrollo GWT.

Tabla IV.2: Matriz de confrontación

		Amenazas									Oportunidades				
		A1	A2	A3	A4	A5	A6	A7	A8	A9	O1	O2	O3	O4	O5
Fortalezas	F1	0	0	0	0	0	0	-	0	-	++	+	+	+	+
	F2	-	0	0	0	0	0	0	0	0	0	++	++	+	+
	F3	0	0	-	0	0	0	-	0	0	++	++	++	+	+
	F4	0	0	0	0	0	0	0	0	0	+	+	+	+	+
	F5	0	0	0	0	0	0	0	0	0	+	++	++	+	++

	F6	0	0	0	0	0	0	0	0	0	+	++	++	+	++	
	F7	0	0	0	0	0	0	0	0	0	+	++	++	+	++	
	F8	0	-	0	0	0	0	0	-	0	0	++	++	++	+	
	F9	0	0	0	0	0	0	0	-	0	0	++	++	++	+	
	F10	0	0	0	0	0	0	0	0	0	+	++	++	++	++	
	F11	0	0	0	0	0	0	0	0	0	0	++	++	+	++	
	F12	0	0	0	0	0	0	0	0	0	0	++	++	++	++	
	F13	0	0	0	-	0	0	0	0	0	0	++	++	+	+	
	F14	-	0	0	0	0	0	0	0	0	+	++	++	++	+	
	F15	0	0	0	-	0	0	0	0	0	0	+	+	+	+	
	F16	0	0	0	0	0	0	0	0	0	++	++	++	++	++	
	Debilidades	D1	-	0	0	0	0	0	-	0	0	0	++	++	++	+
		D2	0	0	0	0	0	0	0	0	0	+	+	+	+	+
		D3	-	-	-	0	0	0	0	0	-	+	+	+	0	0
		D4	0	0	-	0	0	0	0	-	-	0	0	0	0	0
		D5	-	0	-	0	0	0	0	0	0	0	+	+	+	+
D6		-	0	0	-	0	0	0	0	0	++	+	++	+	+	
D7		0	0	0	0	0	0	0	0	0	+	0	0	+	+	
D8		-	0	0	0	0	0	0	0	0	0	0	0	0	0	

4.6.3.3. Elaboración de la matriz FODA modificada

Esta matriz permitirá determinar cuáles son los puntos en donde más se debe poner atención para de esta manera hacer de GWT una plataforma mucho más compacta.

Tabla IV.3: Matriz FODA modificada

ANÁLISIS INTERNO	Fortalezas	ANÁLISIS EXTERNO													
		Amenazas									Oportunidades				
		A1	A2	A3	A4	A5	A6	A7	A8	A9	O1	O2	O3	O4	O5
	F1							-		-					
	F2	-													
	F3			-						-					
	F4														

		F5																
		F6																
		F7																
		F8		-							-							
		F9									-							
		F10																
		F11																
		F12																
		F13					-											
		F14	-															
		F15					-											
		F16																
		Debilidades	D1	-								-						
			D2															
			D3	-	-	-							-					
			D4				-						-	-				
D5	-				-													
D6	-					-												
D7																		
D8	-																	

4.6.4. Diseño de las opciones estratégicas

- Elección de la clasificación de estrategias.
- Despliegue de las estrategias
- Elaboración de la matriz de estrategias

4.6.4.1. Elección de la clasificación de estrategias

Viene sugerida por las estrategias que resultan de la intersección entre filas y columnas, y de acuerdo a esto tenemos las siguientes:

- **Las estrategias de supervivencia (DA):** Destinadas a eludir los efectos que las situaciones del entorno puedan tener sobre los aspectos internos (relacionados con las Debilidades y Amenazas).
- **Las estrategias defensivas (FA):** Pretenden dar respuestas a situaciones del entorno no favorables apoyándose para ello en puntos fuertes de la plataforma en análisis (relacionados con las Fortalezas y Amenazas).
- **Las estrategias de reorientación (DO):** Tratan de aprovechar una situación positiva del entorno para corregir carencias o debilidades internas de la institución (relacionados con las Debilidades y Oportunidades).
- **Las estrategias ofensivas (FO):** Tratan de obtener el máximo provecho de una situación favorable del entorno (relacionados con las Fortalezas y Oportunidades).

Tabla IV.4: Elección de la clasificación de estrategias

		Factores Internos	
		Fortalezas (F)	Debilidades (D)
Factores Externos	Oportunidades (O)	Estrategia Ofensiva (FO) (Maxi – Maxi) Estrategia para maximizar F y O	Estrategia de reordenamiento (DO) (Mini - Maxi) Estrategia para minimizar D y maximizar O.
	Amenazas (A)	Estrategia Ofensiva (FA) (Maxi – Mini) Estrategia para maximizar F y minimizar A.	Estrategia de Supervivencia (DA) (Mini – Mini) Estrategia para minimizar D y A.

4.6.4.2. Despliegue de las estrategias

Se describe cada una de las estrategias que se utilizará en cada una de las relaciones del FODA, así como también la descripción a la que va a ser sujeta.

Tabla IV.5: Despliegue de las estrategias

	Relación	Estrategia	Descripción
E.S. (DA)	D1-A1	ES 1	Promocionar por medio de ejemplos en la web las bondades del GWT frente a otras tecnologías.
	D1-A7	ES 2	Antes de realizar una aplicación, asegurarse que este activo JavaScript en el navegador.
	D3-A1	ES 3	Tratar de trabajar solamente con AJAX para todas las aplicaciones web por ser la tendencia en la actualidad.
	D3-A2	ES 4	Subir a la web más documentación referente a la forma de manejo de los motores de búsqueda.
	D3-A3	ES 5	Hacer conocer sobre la existencia de aplicaciones internacionales como Gmail que ha sido desarrollado con GWT.
	D3-A9	ES 6	Estudiar un poco la documentación expuesta en la web para poder usar de una manera sencilla este nuevo framework.
	D4-A3	ES 7	Para los sitios internacionales, subir la aplicación en servidores que brinden la seguridad posible para evitar ataques de hacker.
	D4-A8	ES 8	Aplicar otros mecanismos de encapsulamiento de clases para contrarrestar la inseguridad que puede brindar al momento de recargar la página.
	D4-A9	ES 9	Diseñar la aplicación mediante tablas para que de esta manera se pueda evitar que se dañe el diseño en una pantalla que no cumpla con las características adecuada
	D5-A1	ES 10	Depurar la primera vez la aplicación en una máquina que disponga de Internet.
	D5-A3	ES 11	En lo posible desarrollar la aplicación dentro de la empresa o contratar un plan de Internet para poder trabajar de una manera cómoda.
	D6-A1	ES 12	Conocer los conceptos básicos de JAVA antes de empezar el trabajo con GWT.
	D6-A4	ES 13	Estudiar todos los libros relacionados a GWT para tener una curva de aprendizaje más adecuada.

	D8-A1	ES 14	Conocer las bondades que se tiene con GWT, sabiendo que al inicio vamos a consumir memoria del navegador pero que luego nos va a servir de mucho
E.D. (FA)	F1-A7	ED 1	Aprovechar al máximo el hecho de instalar el framework GWT, aprovechando los recursos del sistema.
	F1-A9	ED 2	Reducir el tiempo de aprendizaje por la facilidad de no instalar y solo ejecutar.
	F2-A1	ED 3	Con las actualizaciones automáticas rápidamente alcanzará una popularidad en el desarrollo de aplicaciones.
	F3-A3	ED 4	Utilizar al máximo la fortaleza de tener una plataforma independiente para crear sitios de una manera rápida.
	F3-A7	ED 5	Usar un diseñador gráfico en la creación de las aplicaciones.
	F8-A2	ED 6	Subir documentos de la reutilización de componentes en GWT para que sea más popular in entendible para cualquier desarrollador.
	F8-A8	ED 7	Crear componentes reutilizables para el contenido de la presentación y para la lógica.
	F9-A7	ED 8	Utilizar el RPC para acceder a la base de datos por tener métodos primitivos de conexión.
	F13-A4	ED 9	Estudiar más a fondo el uso de los JUnit para que la curva de aprendizaje no sea tan lenta.
	F14-A1	ED 10	Usar la internacionalización en cada una de las aplicaciones web
	F15-A4	ED 11	Estudiar a lo básico JavaScript para obtener la interoperabilidad y control del sistema.

4.6.4.3. Elaboración de la matriz de estrategias

Aquí se escribirá en cada uno de los puntos en los que la plataforma flaqueaba las estrategias necesarias para poder resolver el problema, y mejorar con ello la utilización de GWT.

Tabla IV.6: Matriz de estrategias

		ANALISIS EXTERNO													
		Amenazas									Oportunidades				
		A1	A2	A3	A4	A5	A6	A7	A8	A9	O1	O2	O3	O4	O5
Fortalezas	F1							ED1		ED2					
	F2	ED3 ED1													
	F3			ED4					ED5						
	F4														
	F5														
	F6														
	F7														
	F8		ED6						ED7 ED2						
	F9							ED8							
	F10														
	F11														
	F12														
	F13				ED9										
	F14	ED10 ED 1													
	F15				ED11 ED6										
	F16														
Debilidades	D1	ES1						ES2							
	D2														
	D3	ES3 ES1	ES4 ES2	ES5					ES6						
	D4			ES7 ES1				ES8 ES9							
	D5	ES10 ES1		ES11											
	D6	ES12 ES1			ES13 ES1										
	D7														

		D8	ES14 ES1													
--	--	-----------	-------------	--	--	--	--	--	--	--	--	--	--	--	--	--

4.6.5. Ponderación de las estrategias

En este punto se va a determinar la importancia relativa de las estrategias consideradas, de acuerdo a los siguientes puntos:

- Determinación del peso correspondiente a cada estrategia en cada grupo (número de veces que aparece la estrategia y de su participación porcentual en el total del grupo).
- Valoración de las estrategias en cada grupo

Luego de esto se procederá a realizar la correspondiente jerarquización de Estrategias tanto de Supervivencia como defensivas.

4.6.5.1. Jerarquización de Estrategias de Supervivencia (ES).

Para obtener una buena jerarquización de Estrategias de Supervivencia (ES), se va a realizar el conteo del número de veces que se ha utilizado cada una de las estrategias para luego proceder a la realizar la respectiva jerarquización.

Tabla IV.7: Jerarquía de Estrategias de Supervivencia (ES)

CODIGO	ESTRATEGIA	%
ES1	7	33,33
ES2	2	9,52
ES3	1	4,76
ES4	1	4,76

ES5	1	4,76
ES6	1	4,76
ES7	1	4,76
ES8	1	4,76
ES9	1	4,76
ES10	1	4,76
ES11	1	4,76
ES12	1	4,76
ES13	1	4,76
ES14	1	4,76
TOTAL	21	100

4.6.5.2. Jerarquización de Estrategias Defensivas (ED).

Para obtener una buena jerarquización de Estrategias Defensivas (ED), se va a realizar el conteo del número de veces que se ha utilizado cada una de las estrategias de este tipo para luego proceder a la realizar la respectiva jerarquización.

Tabla IV.8: Jerarquía de Estrategias Defensivas (ED)

CODIGO	ESTRATEGIA	%
ED1	3	20
ED2	2	13,33
ED6	2	13,33
ED3	1	6,67
ED4	1	6,67
ED5	1	6,67
ED7	1	6,67
ED8	1	6,67
ED9	1	6,67
ED10	1	6,67
ED11	1	6,67
TOTAL	15	100

4.6.6. Interpretación de resultados

- El FODA es una herramienta analítica que facilita sistematizar la información que posee la plataforma GWT en el mercado, con fin de definir su capacidad competitiva en un período determinado. Por lo general, es utilizada por los niveles directivos y en este caso por los autores de este trabajo de investigación, reuniendo información de factores externos e internos a efectos de establecer Fortalezas, Oportunidades, Debilidades y Amenazas (FODA).
- Este análisis FODA combina el interior de la plataforma GWT (fortalezas y debilidades) con las fuerzas externas (oportunidades y amenazas) para sacar las diferentes estrategias que permitirán hacer de GWT una herramienta más eficiente.
- Las Estrategias Ofensivas (FO) son aquellas que permitirán maximizar las fortalezas y las oportunidades cuando estas son demasiadas bajas, mientras que las Estrategias de Reordenamiento (DO) son aquellas que se encargan de minimizar las debilidades maximizando las oportunidades.
- En lo referente a las Estrategias Ofensivas (FA) se busca maximizar las fortalezas minimizando las amenazas y por último las Estrategias de Supervivencia (DA) son aquellas que se encargan de minimizar tanto las debilidades y las amenazas de la plataforma GWT para que ésta sea lo más eficiente posible.
- En el análisis del FODA sobre las ventajas y desventajas de la plataforma de desarrollo GWT se observa que no fue necesario establecer estrategias de reordenamiento (DO) porque las oportunidades frente a las debilidades son mayores, ni tampoco fueron

necesario estrategias Ofensivas (FO) puesto que tanto las oportunidades y las fortalezas son positivas obteniendo ++.

- En la jerarquización de las Estrategias de Supervivencia, se tiene que la estrategia número 1 es la que más actúa en la tabla para minimizar los riesgos repitiéndose 7 veces y alcanzando un 33,33% de puntaje, seguida por la segunda estrategia con un 9,52% con un número de repetición de 2, mientras que el resto de estrategias solo actúa una vez por lo que tienen un porcentaje de 4,76%.
- Para la jerarquía de las otras estrategias (Defensivas), se puede observar que la primera estrategia es la que más se repite obteniendo un porcentaje de 20%, mientras que la estrategia 2 y 6 ocupan el segundo lugar con valor de 13,33% y con un menor porcentaje se encuentran las estrategias numero 3, 4 y 5 con un valor de 6,67%.

CAPÍTULO V

DESARROLLO DE LA APLICACIÓN WEB PARA LOS EMPLEADOS DE REFINERÍA ESTATAL DE ESMERALDAS

5.1. Introducción

En el presente capítulo se pone en marcha la parte aplicativa del proyecto, es decir, la instalación, el manual técnico, manual de usuario y de configuración de la aplicación web para el departamento de Recursos Humanos de la Refinería Estatal de Esmeraldas usando la plataforma de desarrollo GWT, en el servidor de aplicaciones web con el que cuenta la empresa.

Se mostrará los pasos necesarios para poder realizar un proyecto desde cero usando GWT, así como también la configuración necesaria que se debe realizar en el IDE de desarrollo Eclipse y la ayuda que podría necesitar cualquier persona que desee usar esta herramienta.

Por último se presenta la documentación de la aplicación web “SISREHU” que consiste en: Documentación Técnica (SRS), Manual de Instalación, Manual de Usuario, Manual de Configuración.

5.2.Instalación de GWT

5.2.1. Prerrequisitos para instalar GWT

Debemos tener instalado las siguientes herramientas:

- Eclipse v3.4
 - Plugin necesario para el desarrollo de la aplicación
 - Cypal.studio.for.gwt-RC5
 - GChart v2.5
 - Plugin para la base de datos: jt400
- Java
 - JDK v1.6.0_13
 - JRE v6
- Glassfish v2.1

5.2.2. Plugin utilizados

- **Cypal.studio.for.gwt-RC5**

Cypal Estudio es un conjunto de plugins que nos permite trabajar con GWT usando el IDE Eclipse como entorno de desarrollo. Actualmente le ayuda en:

- Creación de un módulo

- Creación de un servicio remoto
- Mantener el archivo asincrónica
- Ejecutar su aplicación en modo hosted
- Compilación de su aplicación a Java Script
- Exportando .WAR

- **GChart v2.5**

Un GChart puede representar y mostrar un gráfico de líneas, un gráfico de barras, un gráfico circular, un gráfico de área, o un gráfico que contiene combinaciones arbitrarias de las líneas, barras, circulares y / o zona de curvas.

Este plugin es utilizado para desarrollar cuadros estadísticos para cumplir con los requerimientos de la aplicación.

- **JT400**

El plugin jt400 sirve para la conexión de la base de datos DB2 y la aplicación web, para así extraer los datos necesarios y ser mostrados en el sitio web.

5.2.3. Instalación de GWT en la PC

Lo siguiente es distinto para cada sistema operativo, pero los pasos a seguir en concreto son:

- a. Descargar el software desde esta dirección:

<http://code.google.com/webtoolkit/download.html>.

- b. Descomprimir el software.

- c. Copiar el software descomprimido a la carpeta C:\

- d. Click derecho sobre Mi PC seleccionar Propiedades, la pestaña Opciones avanzadas y seleccionar la Variable de Entorno.

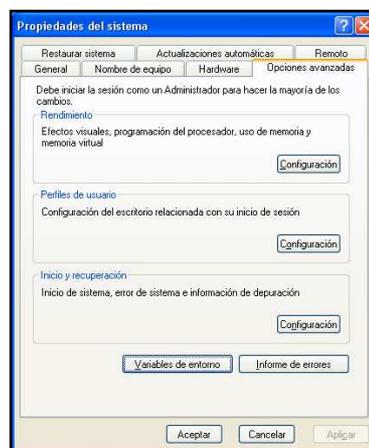


Gráfico V.1: Instalación GWT en la PC (Paso 4)

- e. Damos click en el botón **Nueva** de las Variables del Sistema.

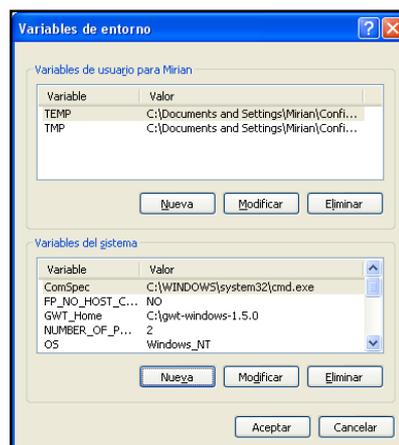


Gráfico V.2: Instalación GWT en la PC (Paso 5)

- f. En Nombre de la variable escribimos **GWT_Home** como lo llamaremos en la instalación y en el Valor de la variable la ubicación de la carpeta copiada en el disco C:\ y luego click en Aceptar.



Gráfico V.3: Instalación GWT en la PC (Paso 6)

- g. Damos click en el botón Aceptar, y Aceptar; concluimos la instalación en el equipo.

5.3. Configuración de GWT en Eclipse

- a. Abrimos el programa Eclipse.
- b. Seleccionamos la opción Windows y damos click en Preferences.

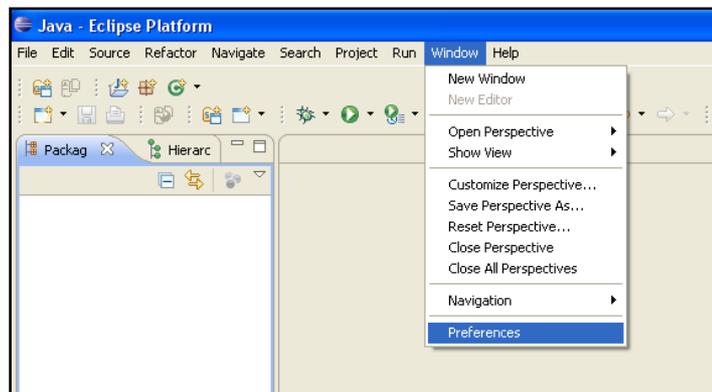


Gráfico V.4: Instalación GWT en Eclipse (Paso 2)

- c. A continuación damos click en Cypal Studio y configuramos los espacios en blanco.

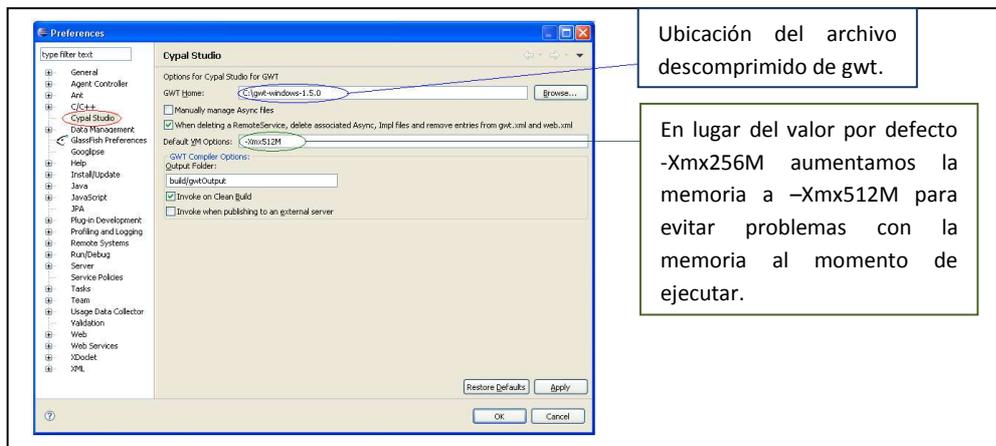


Gráfico V.5: Instalación GWT en Eclipse (Paso 3)

d. Luego click en el **OK** y tenemos configurado GWT para realizar los diferentes proyectos.

5.4. Conexión con Base de Datos

5.4.1. Conexiones con AS/400

Para lograr la correcta conexión con la base de datos DB2 que se encuentra alojada en la Refinería Estatal de Esmeraldas, es necesario conocer algunos conceptos que se muestran a continuación y que son de mucha importancia para el correcto aprendizaje.

5.4.2. JDBC

JDBC es un API de Java para acceder a sistemas de bases de datos, y prácticamente a cualquier tipo de dato tabular. El API JDBC consiste de un conjunto de clases e interfaces que permiten a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea. Al igual que ODBC, la aplicación de Java debe tener acceso a un controlador

(*driver*) JDBC adecuado. Este controlador es el que implementa la funcionalidad de todas las clases de acceso a datos y proporciona la comunicación entre el API JDBC y la base de datos real. De manera muy simple, al usar JDBC se pueden hacer tres cosas:

- Establecer una conexión a una fuente de datos (ej. una base de datos).
- Mandar consultas y sentencias a la fuente de datos.
- Procesar los resultados.

5.4.3. Cargar el controlador JDBC

Para trabajar con el API JDBC se tiene que importar el paquete **java.sql**, tal y como se indica a continuación:

```
import java.sql.*;
```

En este paquete se definen los objetos que proporcionan toda la funcionalidad que se requiere para el acceso a bases de datos.

El siguiente paso después de importar el paquete `java.sql` consiste en cargar el controlador JDBC, es decir un objeto **Driver** específico para la base de datos DB2 y para nuestro caso es el `jt400`.

Hay varias formas de hacerlo, pero la más sencilla es utilizar el método **forName()** de la clase **Class**:

```
Class.forName("Controlador JDBC");
```

Para el caso particular del controlador para DB2, se tiene lo siguiente:

```
Class.forName("com.ibm.as400.access.AS400JDBCdriver").newInstance();
```

Debe tenerse en cuenta que el método estático `forName()` definido por la clase `Class` genera un objeto de la clase especificada. Cualquier controlador JDBC tiene que incluir una parte de iniciación estática que se ejecuta cuando se carga la clase. En cuanto el cargador de clases carga dicha clase, se ejecuta la iniciación estática, que pasa a registrarse como un controlador JDBC en el **DriverManager**, el URL sería:

```
String URL = "jdbc:as400://172.17.16.70/SRIPDATA";
```

Una vez que se ha determinado el URL, se puede establecer una conexión a una base de datos.

El objeto **Connection** es el principal objeto utilizado para proporcionar un vínculo entre las bases de datos y una aplicación en Java. `Connection` proporciona métodos para manejar el procesamiento de transacciones, para crear objetos, ejecutar instrucciones SQL, y para crear objetos para la ejecución de procedimientos almacenados.

Se puede emplear tanto el objeto `Driver` como el objeto `DriverManager` para crear un objeto `Connection`. Se utiliza el método **connect()** para el objeto `Driver`, y el método **getConnection()** para el objeto `DriverManager`.

El objeto `Connection` proporciona una conexión estática a la base de datos. Esto significa que hasta que se llame en forma explícita a su método `close()` para cerrar la conexión o se destruya el objeto `Connection`, la conexión a la base de datos permanecerá activa.

La manera más usual de establecer una conexión a una base de datos y que nosotros hemos usado es invocando el método `getConnection()` de la clase `DriverManager`. A menudo, las bases de datos están protegidas con nombres de usuario (login) y contraseñas (password) para restringir el acceso a las mismas. El método `getConnection()` permite que el nombre de usuario y la contraseña se pasen también como parámetros como veremos a continuación.

```
String USUARIO = "RHESPOCH";  
String PASS = "RHESPOCH";  
Connection conexion = DriverManager.getConnection(URL, USUARIO, PASS);
```

El siguiente código es un ejemplo que va a servir para verificar que podemos establecer una conexión a la base de datos de la refinería.

```
import java.sql.*;  
public class ConexionBD  
{  
    static String BD = "SRIPDATA";  
    static String usuario = "RHESPOCH";  
    static String pass = "RHESPOCH";  
    static String cadena_conexion = "jdbc:as400://172.17.16.70/+\"BD\"";  
    static final String driver = "com.ibm.as400.access.AS400JDBCdriver";  
    public static void main(String[] args) throws Exception {  
        Connection conn = null;
```

```
try{
    Class.forName(driver).newInstance();
    conn = DriverManager.getConnection(cadena_conexion, usuario,
    pass);
    if (conn != null){
        System.out.println("Conexión a base de datos "+url+" ... Ok");
        conn.close();
    }
}
catch(SQLException ex){
    System.out.println(ex);
}
catch(ClassNotFoundException ex){
    System.out.println(ex);
}
}
```

5.5.Desarrollo de un proyecto GWT

5.5.1. Crear un proyecto GWT

Para crear un proyecto con GWT en Eclipse se va a realizar los siguientes pasos:

- a. Crear un proyecto Web Dinámico en Eclipse. Para ello en eclipse vamos al menú principal y selecciona la opción: File -> New -> Project.
- b. Dentro de la carpeta Web, elige Dynamic Web Project como se muestra en la figura y presiona el botón Next.

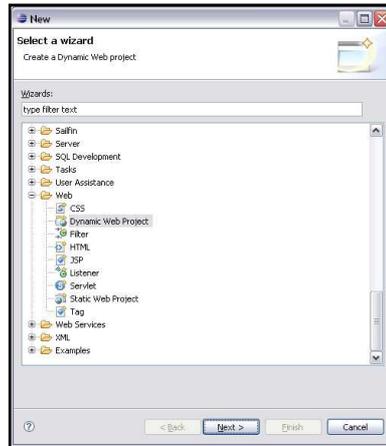


Gráfico V.6: Pantalla para crear un proyecto web dinámico

- c. A continuación se llenará lo siguiente: nombre del proyecto, en Target Runtime, en Dynamic elige la versión 6 de java, en Web Module versión seleccionar 2.4 y en la configuración, selecciona Cypal Studio for GWT.



Gráfico V.7: Llenar datos para crear un proyecto GWT

- d. Luego dar click en Finish, y se termina la primera parte para crear un proyecto.

5.5.2. Configuración de la Librería GWT

- a. Para acceder al build path del proyecto creado anteriormente, hacer click derecho sobre el proyecto en el Project Explorer elegir la opción *Properties*.

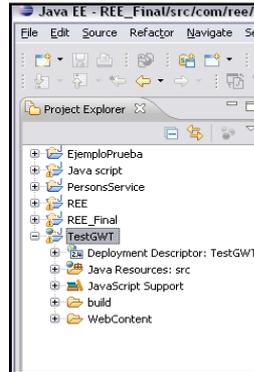


Gráfico V.8: Primera pantalla para configurar la librería GWT

- b. Elegir *Java build Path*, esto llevará a la lista de librerías ya incluidas en el proyecto. Ahí se puede ver en la parte derecha que por defecto Cypal nos incluyó la librería de GWT.

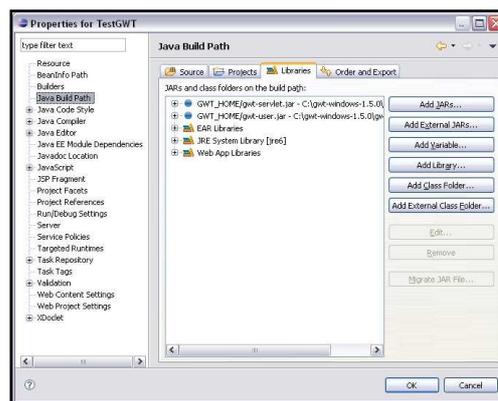


Gráfico V.9: Segunda pantalla para configurar la librería GWT

- c. Dar click en **ok** para terminar.

5.5.3. Creación de un módulo en GWT

- a. Hacer click derecho sobre el proyecto, seleccionar New y elegir Other.
- b. Aquí aparece la ventana para elegir que se quiere crear. Seleccionar dentro de la carpeta de Cypal Studio GWT Module y dar click en Next.

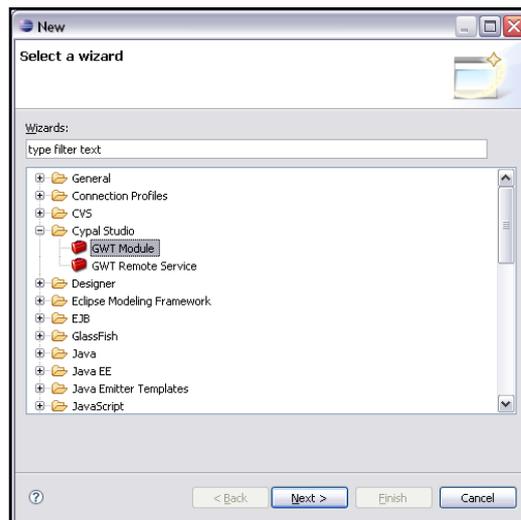


Gráfico V.10: Crear el modulo GWT

- c. A continuación se presenta la ventana para definir el nuevo Modulo de GWT. En el mismo se debe indicar el paquete donde se va a alojar la nueva clase. En este caso *com.ree.ree* y el nombre de la clase que será *TestApp*. Como se puede ver Cypal ya nos sugiere que la clase implemente la interfaz *EntryPoint*, esto es una necesidad de toda clase que corresponda a un nuevo modulo. En GWT debe implementar *EntryPoint*. Así que se deja así, y presionar *Finish* para que Cypal cree la infraestructura necesaria.



Gráfico V.11: Llenar los datos del módulo GWT

- d. Ahora se puede ver lo que Cypal creó por nosotros. Observa que se creó dentro de *Java Resources: src* una jerarquía de paquetes todos bajo el paquete definido previamente *com.ree.ree* que son: *com.ree.ree.client*, *com.ree.ree.server* y el propio *com.ree.ree*.

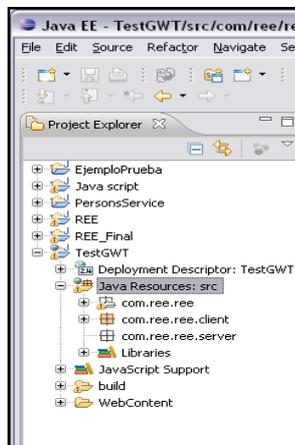


Gráfico V.12: Jerarquía de paquetes

Visualizar lo que se tiene dentro de cada uno de ellos.

- Dentro de *com.ree.ree* se tiene:

- Un directorio *public* con un archivo *Principal.html* esta será la página de inicio de nuestra aplicación web.
 - Un archivo *Principal.gwt.xml* este es el archivo de configuración de GWT para la aplicación.
 - Dentro del paquete *com.ree.ree.client* se tiene:
 - La clase java: *Principal.java* es la que se debe programar, y todas las clases java que se necesite para crear el proyecto GWT.
 - Por último el paquete *com.ree.ree.server* que no se va a utilizar por ahora, pero hay que saber que es aquí donde van a estar los métodos y clases que interactuarán con la base de datos.
- e. Lo primero que se debe hacer es editar el archivo *Principal.gwt.xml* que se encuentra en el paquete *com.ree.ree*. Este archivo contiene la información de configuración de la aplicación GWT. Si lo editamos vemos que contiene una línea que indica con la etiqueta *inherit* la inclusión del Modulo de GWT.
- f. Se debe agregar aquí una nueva línea que indique la inclusión del Modulo de GXT lo que dará acceso a toda la funcionalidad disponible en GXT. Si hizo bien las cosas el archivo *Principal.gwt.xml* debería quedar como sigue:

```
<module>
  <!-- Inherit the core Web Toolkit stuff.      -->
  <inherits name='com.google.gwt.user.User'/>
  <!-- Specify the app entry point class.      -->
  <entry-point class='com.ree.ree.client.Principal'/>
  <inherits name="com.google.gwt.user.theme.standard.Standard"/>
```

```
<!-- <inherits name="com.google.gwt.user.theme.chrome.Chrome"/> -->
<!-- <inherits name="com.google.gwt.user.theme.dark.Dark"/> -->
</module>
```

Ahora se va a escribir las líneas de código Java en la aplicación utilizando las clases del API de GWT para escribir algo muy sencillo.

La aplicación consistirá en:

- Un Botón con el Texto “Ver Mensaje” que cuando se presiona muestra una ventana de información con un mensaje.

El código terminado quedaría algo como esto:

```
public class Principal implements EntryPoint
{
    public void onModuleLoad() {
        this.initialize();
    }
    private void initialize(){
        Button button=new Button("Ver Mensaje");
        button.addSelectionListener(new SelectionListener<ComponentEvent>(){
            @Override
            public void componentSelected(ComponentEvent ce) {
                Info.display("Mensaje", "Este es un mensaje Informativo");
            } });
        RootPanel.get().add(button);
    }
}
```

5.5.4. Ejecución del Código

Si no se tiene ninguna Configuración para Hosted Mode se creará una haciendo lo siguiente:

- a. Hacer click derecho sobre el proyecto, seleccionar la opción *Run As* y elegir *Run Configurations*.
- b. Dar click derecho en la imagen GWT Hosted Mode Application y seleccionar *new*.
- c. Luego completar los campos correspondientes, escribir un nombre al proyecto, después escoger el nombre del proyecto y al Modulo que desea ejecutar.
- d. Luego presione el Botón **Apply** y **Run**, como se muestra a continuación.

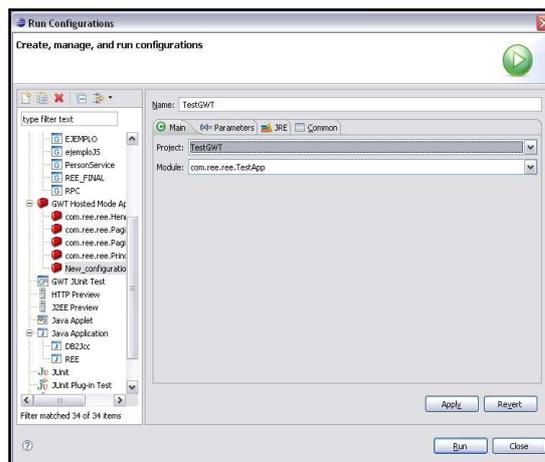


Gráfico V.13: Ejecutar una aplicación GWT con Hosted Mode

Nota: También se puede correr la aplicación directamente en el Browser seleccionando el Botón **Compile** de la ventana del Hosted Mode, el proyecto se compilará automáticamente con el Compilador de GWT a Javascript y se abrirá una instancia del browser por defecto y le mostrará la aplicación dentro de él.

Cuando se ejecuta en el modo hosted, se abren dos ventanas. La primera ventana, mostrada en el siguiente gráfico es llamada **Google Web Toolkit Development Shell / Port 8888**. Esta ventana contendrá los mensajes de errores y log de GWT. Usar la barra de herramientas, se puede abrir un nuevo hosted browser así como ampliar, contraer, y borrar los log en pantalla.



Gráfico V.14: Primera ventana del modo hosted Mode Shell Windows

La segunda ventana, mostrada es el navegador simulado. Como se puede observar, esta contiene los HTML estáticos desde la página principal.html.



Gráfico V.15: Modo hosted simulando el navegador

5.6.Documentación Técnica de la Aplicación Web “SISREHU”

5.6.1. Planificación y Análisis: Documento SRS

5.6.1.1.Introducción

La siguiente documentación técnica basada en la Especificación de Requerimientos de Software (SRS), surge por la necesidad de presentar un conjunto de información necesaria, que ayude a los autores y desarrolladores de la presente aplicación web a analizar y puedan entender todos los requerimientos que los usuarios de la REE desean. De esta forma se desea obtener un documento necesario cuya información se utilizará para el desarrollo completo del software.

Para el desarrollo del SRS se describirá en forma detallada los requerimientos generales de la aplicación web para el departamento de recursos Humanos de la Refinería Estatal de Esmeraldas (REE), el mismo que permitirá el trabajo de forma colectiva (en red), al momento de realizar las diferentes consultas por parte de los empleados (normales y administradores).

Finalmente se logrará alcanzar el objetivo primordial de este capítulo que es identificar y presentar los requerimientos que serán implementados en el Sistema “SISREHU”, de tal forma que se pueda demostrar todo lo que los autores se han propuesto y realizar al final las pruebas correspondientes.

5.6.1.2.Riesgos

5.6.1.2.1. Listado de Riesgos

Tabla V.1: Lista de Riesgos

ID	Descripción de los Riesgos
R1	Nuevos requerimientos por parte del personal de Recursos Humanos de la Refinería Estatal de Esmeraldas.
R2	Cambio de nombres tablas o campos de la Base de Datos en la matriz Petroecuador.
R3	Falta de experiencia de los desarrolladores con respecto a la tecnología que se está utilizando (Framework RIA GWT y JAVA).
R4	Enfermedad de uno de los desarrolladores de la Aplicación.
R5	Cambio de jefe en los departamentos de Sistemas de la matriz Petroecuador y Refinería Estatal de Esmeraldas.
R6	Entrega de la Aplicación fuera del tiempo establecido.
R7	Falta de tiempo por parte del director de Tesis.
R8	Cambio de miembros del tribunal de Tesis.

5.6.1.2.2. Tabla de Valores

A continuación se detallarán los valores con lo que se calificará a los riesgos.

Tabla V.2: Determinación de la Probabilidad

RANGO DE PROBABILIDAD	DESCRIPCIÓN	VALOR
1% - 33%	Baja	1
34% - 67%	Media	2
68% - 99%	Alta	3

Tabla V.3: Determinación del Impacto

IMPACTO	RETRASO	IMPACTO TÉCNICO	IMPACTO EN COSTO	VALOR
Bajo	1 semana	Ligero impacto en el desarrollo del proyecto	< 1%	1
Moderado	2 semana	Moderado efecto en el desarrollo del proyecto	< 5%	2
Alto	1 mes	Severo efecto en el desarrollo del proyecto	< 10%	3
Crítico	> 1 mes	Proyecto que no puede ser culminado	>20%	4

Tabla V.4: Determinación de la Exposición del Riesgo

EXPOSICIÓN	VALOR	COLOR
Baja	1 o 2	Verde
Media	3 o 4	Amarillo
Alta	Mayor a 6	Rojo

Tabla V.5: Equivalencia de riesgos

IMPACTO/PROBABILIDAD	BAJO =1	MODERADO =2	ALTO =3	CRÍTICO =4
Alto = 3	3	6	9	12
Media = 2	2	4	6	8
Baja = 1	1	2	3	4

5.6.1.2.3. Análisis de Riesgos

Tabla V.6: Análisis de Riesgo

ID	PROBABILIDAD			IMPACTO		EXPOSICIÓN AL RIESGO	
	%	Valor	Probabilidad	Valor	Impacto	Valor	Exposición
R1	50	2	Media	2	Media	4	Media
R2	70	3	Alta	2	Media	6	Alta
R3	30	1	Baja	1	Baja	1	Baja
R4	50	2	Media	1	Baja	2	Baja
R5	40	2	Media	1	Baja	2	Baja
R6	50	2	Media	2	Media	4	Media
R7	50	2	Media	2	Media	4	Media
R8	35	2	Media	2	Media	4	Media

5.6.1.2.4. Resultados

Tabla V.7: Resultado del Análisis de Riesgo

ID	EXPOSICION
R2	6
R1	4
R6	4
R7	4
R8	4
R3	1
R4	2
R5	2

Luego de calificar a cada uno de los riesgos mencionados anteriormente, se deduce que el Riesgo es de exposición Media y Baja.

5.6.1.3.Requerimientos Funcionales

5.6.1.3.1. Actores

Tabla V.7: Actores de Caso de Uso

ID	Nombres	Descripción
A1	Usuario_Invitado	Usuario invitado ingresa a la página y puede ver la información general de la Refinería Estatal de Esmeraldas.
A2	Usuario	Empleado de Refinería Estatal de Esmeraldas que visita la página y para ver la información general de la empresa y sus datos personales.
A3	Usuario_Administrador	Empleado de Refinería Estatal de Esmeraldas que puede visualizar toda la información de la empresa, datos personales y estadísticas.

5.6.1.3.2. Diagramas de Casos de Uso

- Diagrama de Caso de Uso General

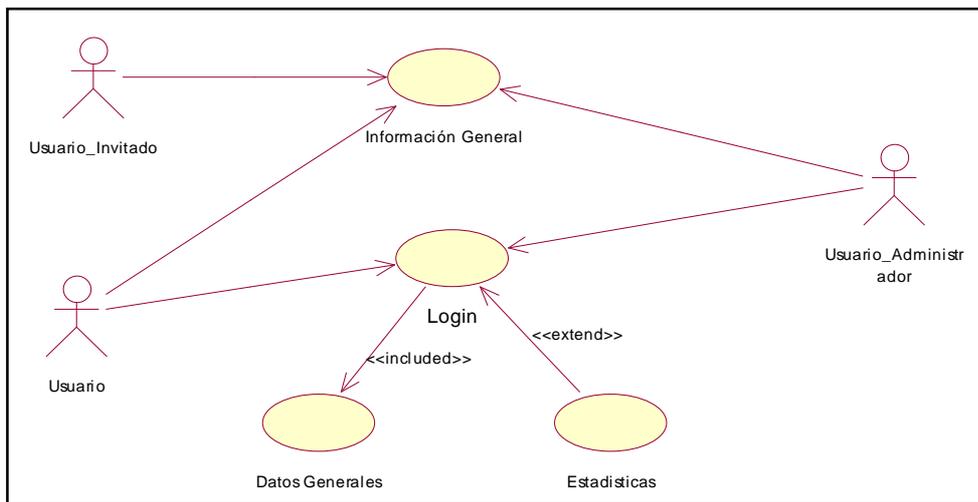


Gráfico V.16: Caso de Uso General

- Diagrama de Caso de Uso del Usuario_Invitado

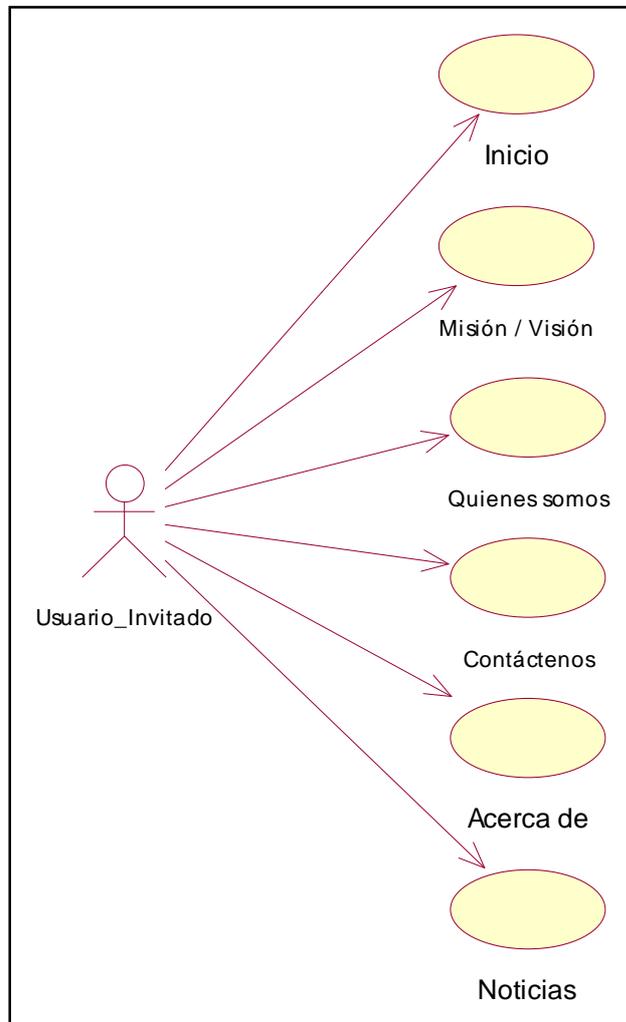


Gráfico V.17: Caso de Uso del Usuario_Invitado

- Diagrama de Caso de Uso del Usuario

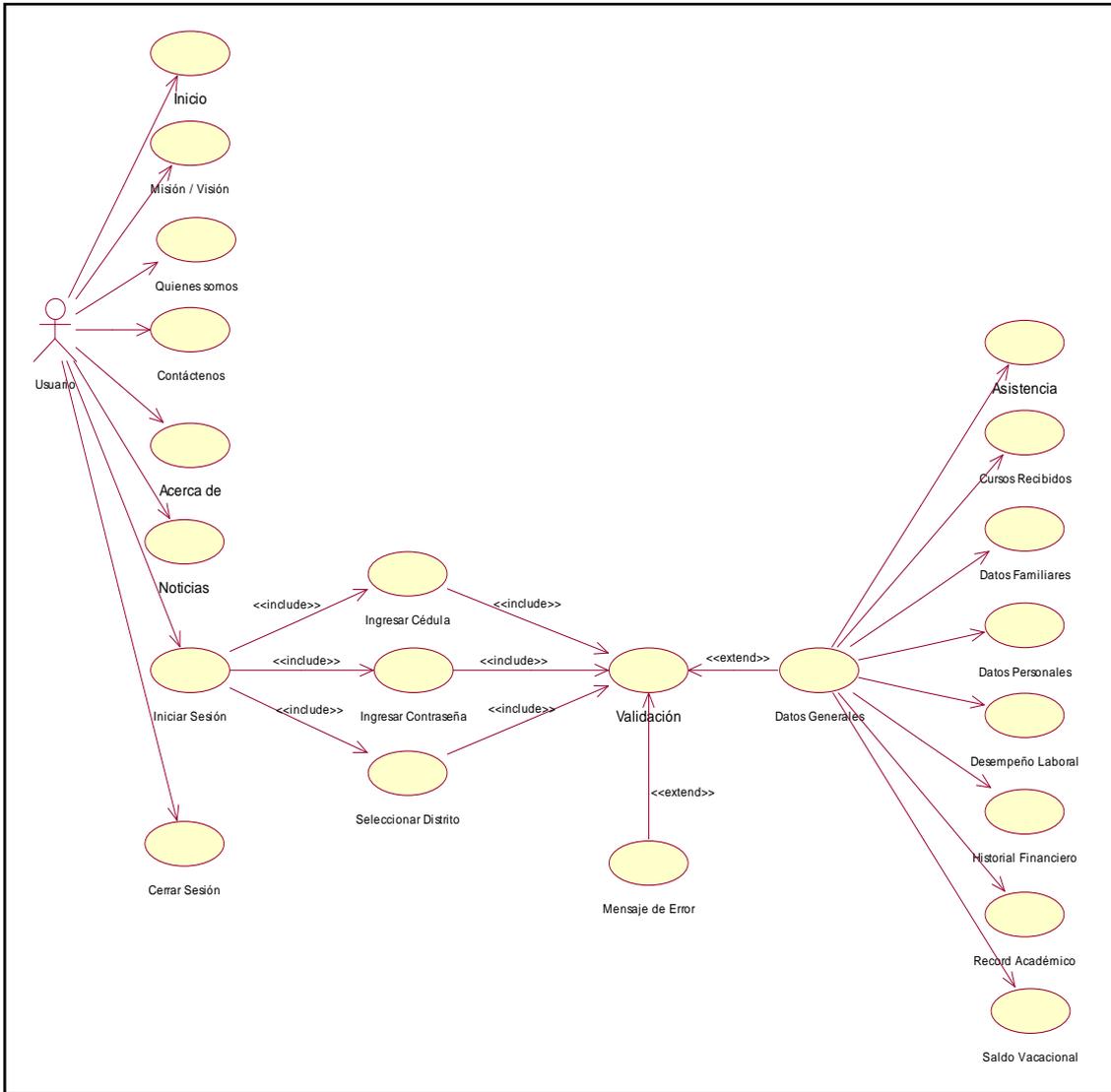


Gráfico V.18: Caso de Uso del Usuario

- Diagrama de Caso de Uso del Usuario_Administrador

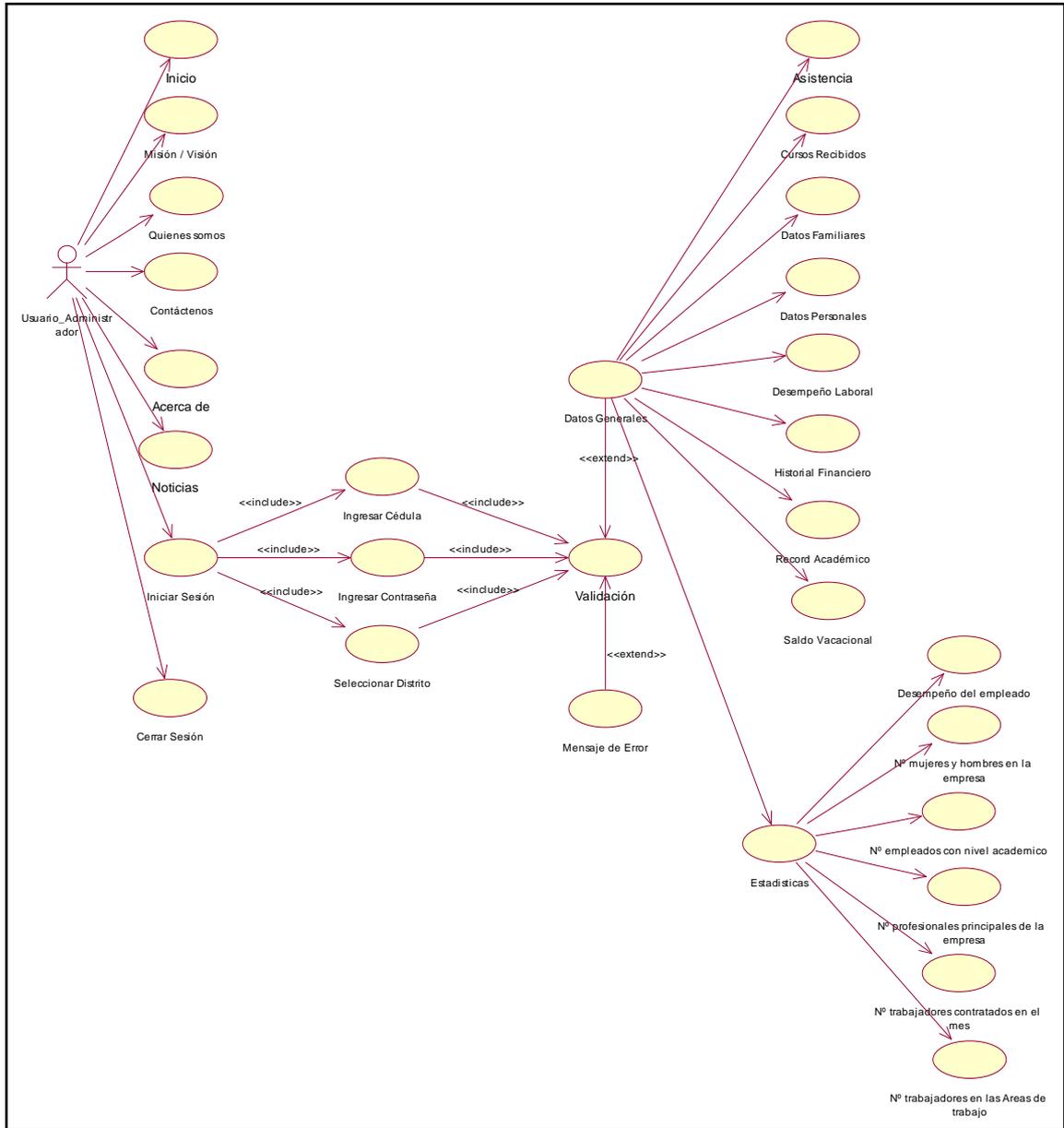


Gráfico V.19: Caso de Uso del Usuario_Administrador

5.6.1.4. Casos de Uso

5.6.1.4.1. Caso de Uso: Información General

Tabla V.9: Caso de Uso de Información General

Identificador Caso de Uso	CU-1	
Nombre del Caso de Uso	Información general	
Actores	USUARIO_INVITADO	
Propósito	Consultar la información general de REE.	
Visión General	Muestra la información general de REE.	
Tipo	Primario	
Referencias		
Curso típico de eventos		
Acción del Actor	Respuesta del Sistema	
1. Seleccionar una opción en el menú principal (inicio, misión/visión, quienes somos, acerca de, noticias).	2. El sistema genera la consulta.	
	3. Muestra la información.	

5.6.1.4.2. Caso de Uso: Iniciar Sesión

Tabla V.10: Caso de Uso de Iniciar Sesión

Identificador Caso de Uso	CU-2
Nombre del Caso de Uso	Iniciar sesión
Actores	USUARIO USUARIO_ADMINISTRADOR
Propósito	Ingresar al módulo de consultas generales y estadísticas
Visión General	El usuario ingresa al sitio web, ingresa a consultar sus datos generales y estadísticas para esto el sistema le solicita la cédula de identidad, su password y que seleccione el distrito al que pertenece, el usuario ingresa

	todos sus datos, el sistema verifica los datos e ingresa al módulo de consultas generales y estadísticas.	
Tipo	Primario	
Referencias		
Curso típico de eventos		
Acción del Actor	Respuesta del Sistema	
1. Ingresa al sitio web	2. Presenta la página de inicio	
3. Ingresa a consultar	4. Solicita los datos, cédula de identidad y al distrito que pertenece el usuario.	
5. Ingresa los datos	6. El sistema valida los datos. 7. Ingresa al módulo de consultas y estadísticas.	
Cursos Alternativos		
Línea 7: Los datos ingresados no son correctos, muestra un mensaje de error y borra los datos ingresados.		

5.6.1.4.3. Caso de Uso: Cerrar Sesión

Tabla V.11: Caso de Uso de Cerrar Sesión

Identificador Caso de Uso	CU-3
Nombre del Caso de Uso	Cerrar sesión
Actores	USUARIO USUARIO_ADMINISTRADOR
Propósito	Cerrar la sesión antes de cerrar el sistema.
Visión General	El usuario, una vez que se haya iniciado sesión es necesario cerrar sesión antes de abandonar el sitio web.
Tipo	Primario
Referencias	

Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita cerrar la sesión	2. El sistema cierra la sesión.

5.6.1.4.4. Caso de Uso: Datos Generales (Asistencia, Cursos Recibidos, Datos Familiares, Datos Personales, Historial Financiero, Record Académico y Saldo Vacacional)

Tabla V.12: Caso de Uso de Datos Generales

Identificador Caso de Uso	CU-4
Nombre del Caso de Uso	Datos generales
Actores	USUARIO USUARIO_ADMINISTRADOR
Propósito	Consultar los datos generales del empleado.
Visión General	Muestra la asistencia, cursos recibidos, datos familiares, datos personales, historial financiero, record académico y saldo vacacional del empleado.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Seleccionar una de las consultas de los datos generales (asistencia, cursos recibidos, datos familiares, datos personales, historial financiero, record académico y saldo vacacional) del empleado.	2. El sistema genera la consulta.
	3. Muestra los datos de la consulta.
L3: En caso de que el empleado no tenga datos ingresados, el sistema mostrará un mensaje.	

5.6.1.4.5. Caso de Uso: Desempeño Laboral

Tabla V.13: Caso de Uso de Desempeño Laboral

Identificador Caso de Uso	CU-5
Nombre del Caso de Uso	Desempeño laboral
Actores	USUARIO USUARIO_ADMINISTRADOR
Propósito	Consultar el desempeño laboral del empleado.
Visión General	Muestra el desempeño laboral de los empleados en un periodo determinado.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Seleccionar la consulta de desempeño laboral del empleado.	3. El sistema genera la consulta.
2. Selecciona el año y periodo.	4. Muestra los datos del desempeño laboral del empleado.
L4: En caso de que el empleado no tenga datos ingresados, el sistema mostrará un mensaje.	

5.6.1.4.6. Caso de Uso: Estadísticas

Tabla V.14: Caso de Uso de Estadísticas

Identificador Caso de Uso	CU-6
Nombre del Caso de Uso	Estadísticas
Actores	USUARIO_ADMINISTRADOR
Propósito	Consultar las estadísticas de REE.
Visión General	Muestra las barras estadísticas.
Tipo	Primario

Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la consulta estadísticas.	2. El sistema genera la consulta.
3. Seleccionar una opción del menú de las estadísticas (desempeño de los empleados, número de mujeres y hombres de la empresa, número de empleados con nivel académico, número de profesionales principales de la empresa, número de trabajadores contratados en el mes, número de trabajadores en las diferentes áreas de trabajo en la empresa).	4. El sistema genera la estadística y luego muestra las barras estadísticas.

5.6.1.5. Requerimientos Detallados

Para un mejor entendimiento de los requerimientos funcionales tanto para la Refinería Estatal de Esmeraldas y los desarrolladores se dividirá en módulos como se detalla a continuación:

Modulo 1: INFORMATIVO

Modulo 2: NOTICIAS

Modulo 3: AUTENTICACIÓN

Modulo 4: CONSULTAS GENERALES

Modulo 5: ESTADISTICAS

5.6.1.5.1. Modulo 1: INFORMATIVO

Req. 1: El sitio web debe mostrar el Inicio es decir la historia de REE.

Req. 2: El sitio web debe mostrar la Misión/Visión de REE.

Req. 3: El sitio web debe mostrar Quienes somos de REE.

Req. 4: El sitio web debe mostrar Acerca de... de REE.

5.6.1.5.2. Modulo 2: NOTICIAS

Req. 5: El sitio web debe mostrar las noticias sobre los distintos departamentos de REE.

5.6.1.5.3. Modulo 3: AUTENTICACIÓN

Req. 6: El sitio web debe permitir al usuario autenticarse.

Req. 7: El sitio web debe permitir cerrar sesión.

5.6.1.5.4. Modulo 4: CONSULTAS GENERALES

Req. 8: El sitio web debe mostrar la asistencia del empleado de REE.

Req. 9: El sitio web debe mostrar los cursos recibidos en el tiempo laborable en la REE.

Req. 10: El sitio web debe mostrar los datos familiares del empleado de REE.

Req. 11: El sitio web debe mostrar los datos personales del empleado de REE.

Req. 12: El sitio web debe mostrar el desempeño laboral del empleado de REE en un periodo determinado.

Req. 13: El sitio web debe mostrar el historial financiero del empleado de REE.

Req. 14: El sitio web debe mostrar el record académico del empleado de REE.

Req. 15: El sitio web debe mostrar el saldo vacacional del empleado de REE.

5.6.1.5.5. Modulo 5: ESTADISTICAS

Req. 16: El sitio web debe mostrar en barras estadísticas el desempeño laboral en un periodo determinado de los empleados de REE.

Req. 17: El sitio web debe mostrar en barras estadísticas el número de mujeres y hombres de REE.

Req. 18: El sitio web debe mostrar en barras estadísticas el número de empleados con nivel académico (primario, secundario, universitario).

Req. 19: El sitio web debe mostrar en barras estadísticas el número de profesionales principales (Ing. Químico, Ing. Mecánico, Ing. Sistemas, etc.) de REE.

Req. 20: El sitio web debe mostrar en barras estadísticas el número de trabajadores contratados en REE en un determinado mes.

Req. 21: El sitio web debe mostrar en barras estadísticas el número de trabajadores en las diferentes áreas de trabajo de la REE.

5.6.1.6. Requerimientos del Sistema

Tabla V.15: Requerimientos del Sistema

Listado de Requerimientos de Sistema	
Id	Descripción
RS-1	El tiempo de respuesta a cualquier petición no debe superar los 5 segundos.
RS-2	El sistema debe soportar a por lo menos 100 usuarios concurrentes trabajando con la aplicación.
RS-3	La aplicación deberá poder ser accesible las 24 horas del día, los 365 días del año.
RS-4	La velocidad de respuesta dependerá también del ancho de banda disponible en la REE para salir al internet.
RS-5	El tiempo de respuesta del sitio web dependerá del hardware que posee la REE.

5.6.1.7. Diccionario de Datos

Tabla V.16: Diccionario de Datos

TABLA	ATRIBUTOS	TIPO DE DATO	DESCRIPCIÓN
ADMINWEB (Administradores)	CEDULA	varchar(10)	Cédula del empleado administrador
	NOMBRE	varchar(35)	Nombre del empleado
RHGDAPT (Datos personales)	CEDULA	varchar(10)	Cédula del empleado
	NOMBRE	varchar(35)	Nombre del empleado
	FECNAC	varchar(8)	Fecha de nacimiento
	NUMEMP	varchar(5)	Número del empleado
	ANOCEP	Integer	Años en petroecuador
	MESCEP	Integer	Mes en petroecuador
	ANOOTI	Integer	Años en otras instituciones
	MESOTI	Integer	Mes en otras instituciones
	SUEEMP	Integer	Sueldo del empleado
	NIVEMP	varchar(2)	Nivel del empleado
	CARFAM	Integer	Cargo del empleado
	FECCRE	varchar(8)	Fecha de creación del empleado
	SEXEMP	varchar(1)	Sexo del empleado
	AREATR	varchar(3)	Área de trabajo
	APEPAT	varchar(20)	Apellido paterno
	APEMAT	varchar(10)	Apellido materno
	PRINOM	varchar(20)	Primer nombre
	SEGNOM	varchar(20)	Segundo nombre
	INDHOR	varchar(1)	Código de horario
	CODLUG	varchar(2)	Código lugar de trabajo
	UNIOPE	varchar(2)	Unidad operativa
	DIATRA	Integer	Días trabajados
	ESTADO	varchar(1)	Estado del empleado
CLAEMP	varchar(10)	Clave del empleado	
RHGESTTC (Estado del empleado)	CEDULA	varchar(10)	Cédula del empleado
	ESTADO	varchar(1)	Estado del empleado
	NOMEST	varchar(30)	Nombre del estado
PSPERMIS (Permisos)	CEDULA	varchar(10)	Cédula del empleado
	NUMEMP	varchar(5)	Nombre del empleado
	AUTPER	varchar(50)	Autorizado por
	CODPER	varchar(50)	Código del permiso
	FDESPE	varchar(8)	Fecha de inicio del permiso
	FHASPE	varchar(8)	Fecha final de permiso
	HDESPE	varchar(2)	Hora desde inicio del permiso
	HHASPE	varchar(2)	Hora hasta fin del permiso
EVAPERTP (Evaluación por)	CEDEVO	varchar(10)	Cédula del evaluado
	CEDEVR	varchar(10)	Cédula del evaluador

factores)	ANOEVA	varchar(4)	Año de la evaluación
	SEMEVA	varchar(2)	Semestre de la evaluación
	PUNCON	Integer	Puntaje de conocimiento
	PUNCAL	Integer	Puntaje de calidad de trabajo
	PUNINI	Integer	Puntaje de iniciativa
	PUNINT	Integer	Puntaje de interés y participación
	PUNADA	Integer	Puntaje de adaptación
	PUNCOM	Integer	Puntaje de conocimiento
	PUNDIS	Integer	Puntaje de disciplina
	PUNREH	Integer	Puntaje de relaciones humanas
	PUNafa	Integer	Puntaje de afán de superación
	PUNORG	Integer	Puntaje de organización
	PUNSEG	Integer	Puntaje de seguridad
	FECEVA	varchar(8)	Fecha de evaluación
	FILIAL	varchar(2)	Código de filial
	POREVA	Integer	Porcentaje de la evaluación
	PUNRES	Integer	Puntaje de seguridad
RCADIHTH (Capacitación)	ORDINA	varchar(2)	Numero ordinal
	CEDULA	varchar(10)	Cédula del empleado
	TEMDIP	varchar(50)	Tema del diploma
	RESDIP	varchar(50)	Responsable del diploma
	DURDIP	varchar(3)	Duración del diploma
	PAIDIP	varchar(15)	País del diploma
	ANOMES	varchar(6)	Año y mes del diploma
	NOMDIP	varchar(50)	Nombre del diploma
	CIUDIP	varchar(4)	Ciudad del diploma
RHTPERTD (Fechas de Ingreso)	CEDULA	varchar(10)	Cédula del empleado
	ORDINA	varchar(2)	Ordinal
	FECING	varchar(8)	Fecha de ingreso del empleado
	FECsAL	varchar(8)	Fecha de salida del empleado
	ESTING	varchar(1)	Estado del ingreso del empleado
RHRDAFTD (Datos familiares)	CEDULA	varchar(10)	Cédula del empleado
	ORDDAF	varchar(2)	Ordinal del familiar
	NOMDAF	varchar(35)	Nombre del familiar
	PARFAM	varchar(1)	Parentesco del familiar
	FENFEM	varchar(8)	Fecha de nacimiento
	CARSUF	varchar(1)	Número de cargas familiares
RHRSVTP (SalDOS de vacaciones)	CEDULA	varchar(10)	Cédula del empleado
	ANODES	varchar(4)	Año desde las vacaciones
	MESDES	varchar(2)	Mes desde las vacaciones
	PAGDOR	varchar(2)	Número de pagador
	ANOHAS	varchar(4)	Año final de las vacaciones
	MESHAS	varchar(2)	Mes final de las vacaciones

	DOFCOL	varchar(1)	Relación laboral
	CONDIC	varchar(1)	Condición (nombramiento, contrato, etc.)
	DEPEND	varchar(13)	Dependencia
	DIAVAC	Integer	Días acreditado en el periodo
	DIAADI	Integer	Días adicionales acreditados
	DIAAOT	Integer	Días adicionales de otras instituciones
	SALVAC	Integer	Salario días acreditado en el periodo
	SALADI	Integer	Salario días acreditados
	SALAOT	Integer	Otros salarios adicionales
	INDP12	varchar(1)	Indicador pago docena
	INDPLI	varchar(1)	Indicador pago liquidación
	INDPAD	varchar(1)	Indicador pago adicional
	ANOPAG	varchar(4)	Año de pago
	MESPAG	varchar(2)	Mes de pago
	FILIAL	varchar(2)	Filial
RESHIETH (Record académico)	ORDINA	varchar(2)	Número ordinal
	CEDULA	varchar(10)	Cédula del empleado
	ANOAPR	Integer	Años aprobados
	SEMAPR	Integer	Semestres aprobados
	ESTADO	varchar(1)	Estado del empleado
	NOMTIT	varchar(50)	Nombre del título
	ANOBT	Integer	Año de obtención del título
	CIUTIT	varchar(4)	Ciudad de obtención del título
RHGCIUTC (Registro de ciudades)	NIVTIT	varchar(1)	Nivel del título
	CODCIU	varchar(4)	Código de ciudad
RHGDISTP (Distributivos de puestos)	NOMCIU	varchar(20)	Nombre de la ciudad
	NUMPUE	varchar(4)	Número del puesto
	CEDULA	varchar(10)	Cédula del empleado
	PAGDOR	varchar(2)	Código del pagador
	CODDEP	varchar(13)	Código de dependencia
	CARFUN	varchar(9)	Cargo funcional del empleado
	DOFCOL	varchar(1)	Relación laboral del empleado
	FILIAL	varchar(2)	Filial a la que pertenece el empleado
RHGDEPTC (Registro de dependencias)	CONDIC	varchar(1)	Condición (nombramiento, contrato..)
	CODDEP	varchar(13)	Código de dependencia
	FILIAL	varchar(2)	Filial a la que pertenece
RHRHINTH (Historial Financiero)	NOMDEP	varchar(50)	Nombre de la dependencia
	CEDULA	varchar(10)	Cédula del empleado
	ANOPRO	varchar(4)	Año de proceso
	MESPRO	varchar(2)	Mes de proceso
	PAGDOR	varchar(2)	Código del pagador

	CONDIC	varchar(1)	Condición (nombramiento, contrato..)
	ESTADO	varchar(1)	Estado del empleado
	CARFUN	varchar(9)	Cargo funcional
	CODDEP	varchar(13)	Código de dependencia
	FILIAL	varchar(2)	Filial a la que pertenece
	TOTING	Integer	Total de ingresos
	OTRDES	Integer	Otros descuentos
	FONRES	Integer	Fondo de reserva
	DECTER	Integer	Décimo tercero
	DECCUA	Integer	Décimo cuarto
	BONJEF	Integer	Bono de jefatura
	DOFCOL	varchar(1)	Relación laboral
	UNIOPE	varchar(2)	Unidad operativa
	TOTDES	Integer	Total de descuentos
RHGPAGTP (Pagadores de rol)	PAGDOR	varchar(2)	Código de pagador
	FILIAL	varchar(2)	Filial a la que pertenece el empleado
	NOMPAG	varchar(20)	Nombre del pagador
RHECARTC (Cargos funcionales)	CARFUN	varchar(9)	Cargo funcional del trabajador
	FILIAL	varchar(2)	Filial a la que pertenece el empleado
	DESCAR	varchar(48)	Nombre del cargo
	NIVCAR	varchar(2)	Nivel del cargo
RHRUNIOPE (Unidades Operativas)	UNIOPE	varchar(2)	Unidad operativa
	NOMUNI	varchar(30)	Nombre de la unidad operativa
RHICONTC (Registro de condiciones)	CONDIC	varchar(1)	Condición (nombramiento, contrato..)
	NOMCON	varchar(30)	Nombre de la condición
ILCARGOS (Cargos de los empleados)	CODCAR	varchar(9)	Código del cargo
	DESPUE	varchar(35)	Descripción del puesto
	FILIAL	varchar(2)	Filial a la que pertenece el empleado
PSYAANAS (Asistencias)	FENYAA	varchar(8)	Fecha entrada timbrada
	HENYAA	varchar(6)	Hora de entrada de la timbrada
	MENYAA	varchar(2)	Minutos de entrada de la timbrada
	FSAYAA	varchar(8)	Fecha de la timbrada de salida
	HSAYAA	varchar(2)	Hora de la salida en la timbrada
	MSAYAA	varchar(2)	Minutos de la salida en la timbrada

5.6.2. Diseño

5.6.2.1. Diseño de la Base de Datos de la Aplicación Web “SISREHU”

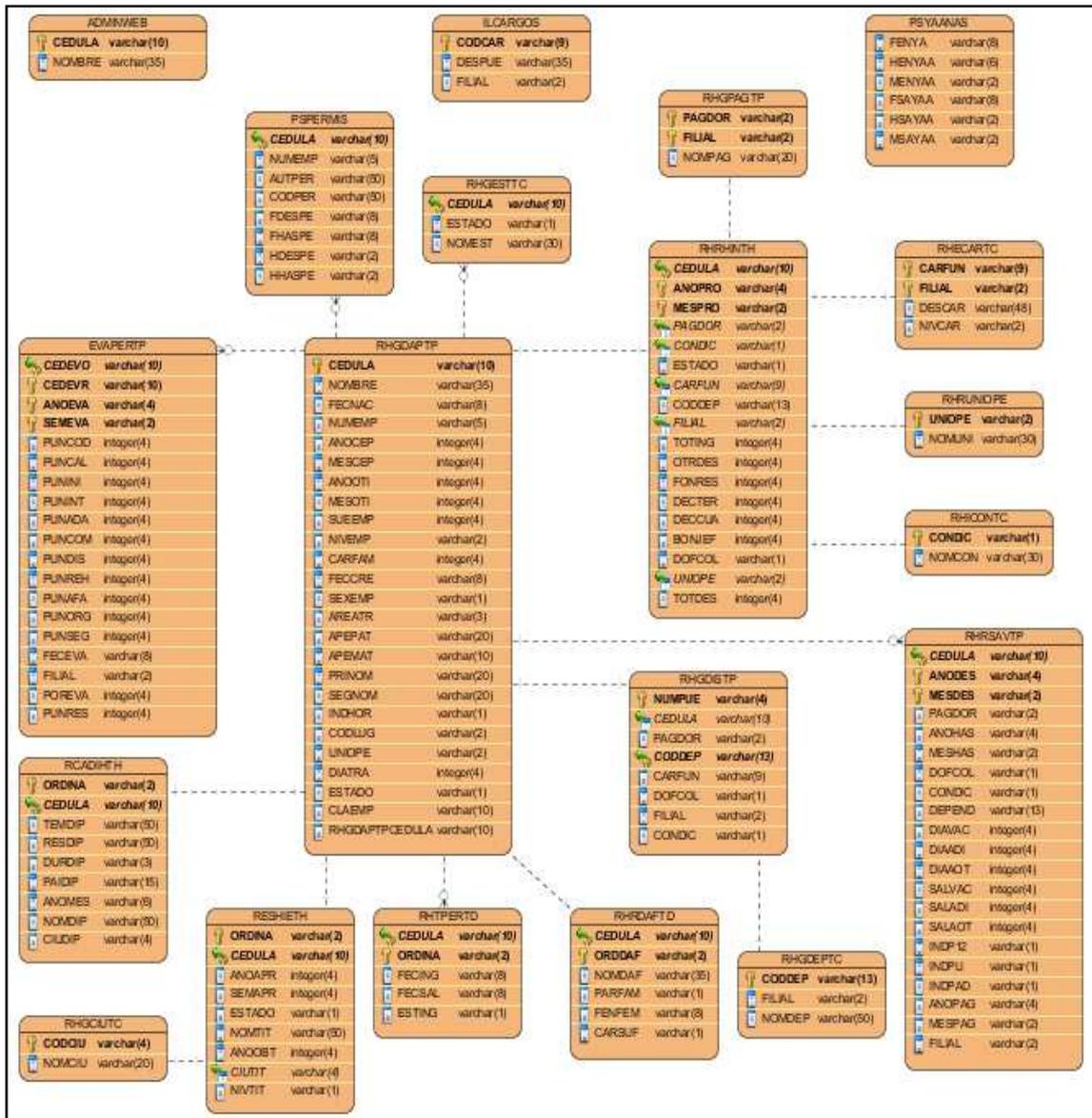


Gráfico V.20: Diseño de la Base de Datos

5.6.2.2. Diagrama de Clases de la Aplicación Web “SISREHU”

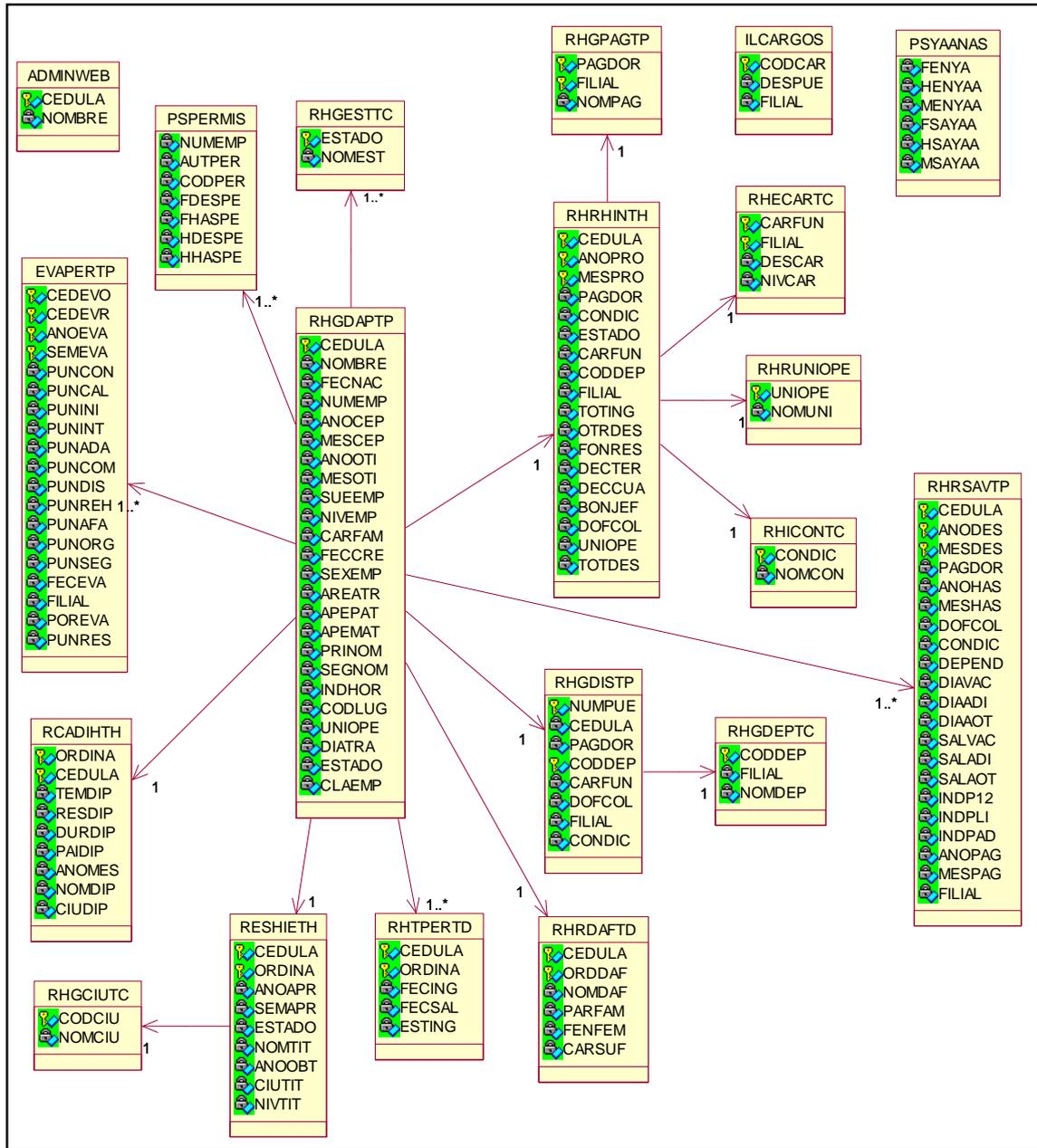


Gráfico V.21: Diagrama de Clases

5.6.2.3. Diagrama de Despliegue de la Aplicación Web “SISREHU”

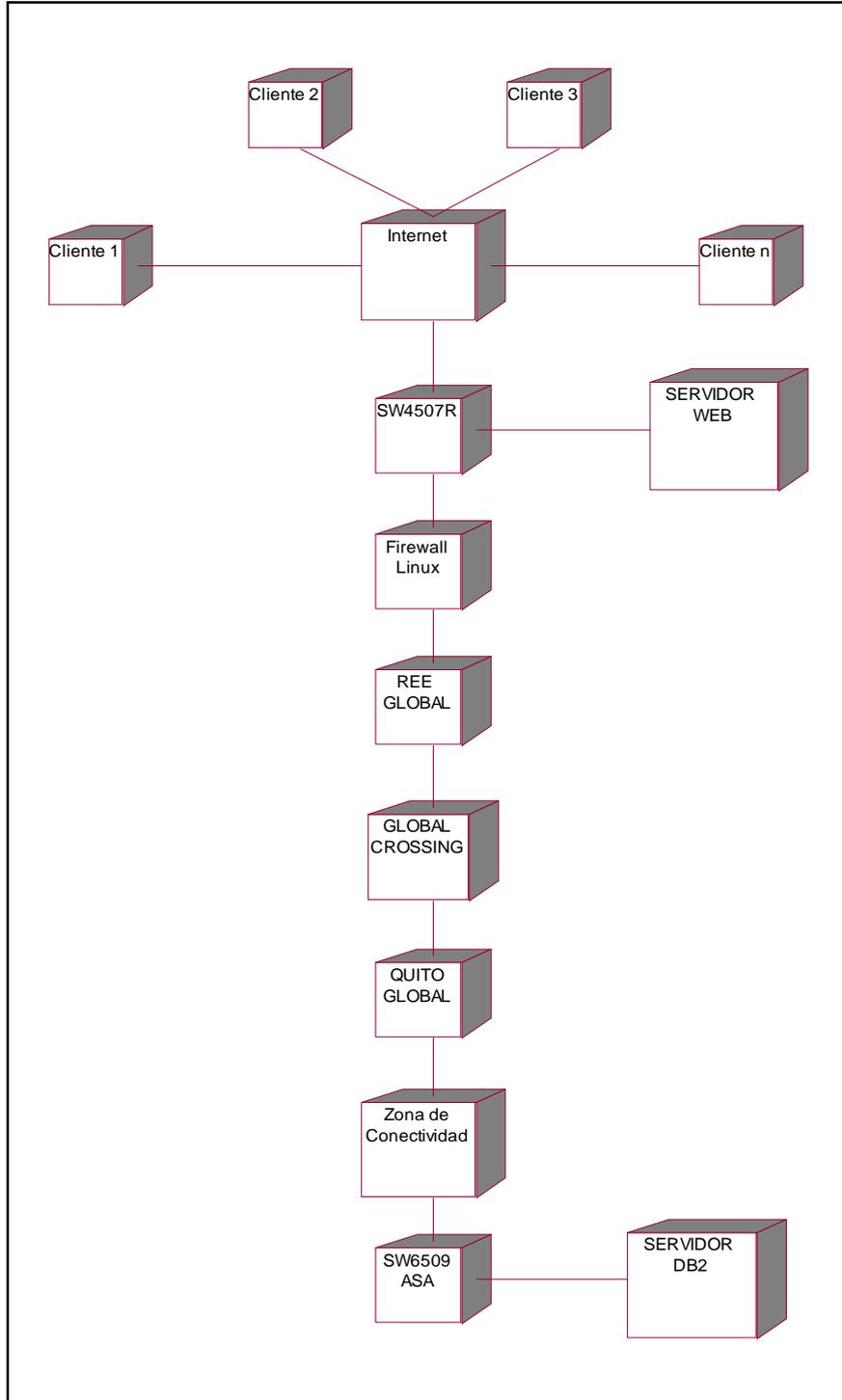


Gráfico V.22: Diagrama de Despliegue

5.6.2.4. Diagrama de Componentes de la Aplicación Web “SISREHU”

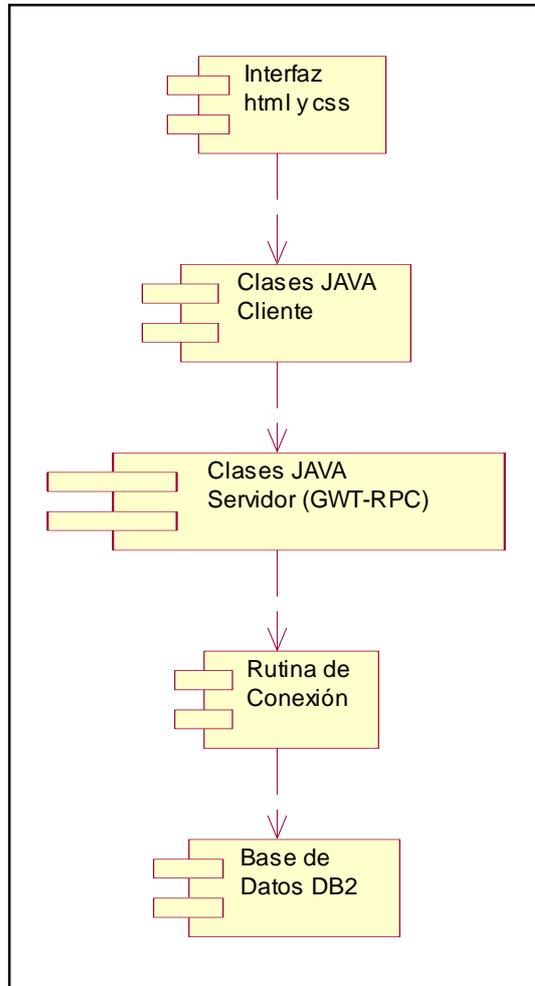


Gráfico V.23: Diagrama de Componentes

5.6.3. Pruebas y Validaciones

Una vez finalizada la aplicación web para el Departamento de Recursos Humanos de la Refinería Estatal de Esmeraldas se procedió a realizar las respectivas pruebas y validaciones en cada uno de los procesos que realiza teniendo como resultado lo siguiente:

5.6.3.1. Pruebas

Para demostrar este punto se procedió a navegar en la aplicación iniciando por:

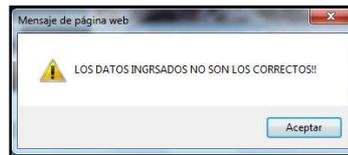
- La autenticación de usuarios, la misma que se ha realizado para los dos tipos de usuarios que va a tener la aplicación mediante su cédula, clave y filial a la que pertenece, el primero de los usuarios es el administrador el cual tendrá todos los permisos, mientras que el segundo usuario luego de autenticarse podrá realizar todas las consultas excepto las estadísticas.
- Se comprobó que los datos obtenidos de las diferentes consultas tales como de: asistencia, cursos recibidos del empleado, datos personales, datos de los familiares, desempeño obtenido, las diferentes estadísticas, récord académico, historial financiero y saldo vacacional, han sido los correctos demostrando así mediante estas pruebas que la aplicación web funciona correctamente.
- El cierre de sesión en una aplicación es de suma importancia para la seguridad que esta pueda brindar al usuario, y en este caso se comprobó copiando la URL una vez que el usuario se ha registrado y pegándola en otra ventana del navegador donde se puede observar que la aplicación automáticamente envía al usuario a autenticarse.

5.6.3.2. Validaciones

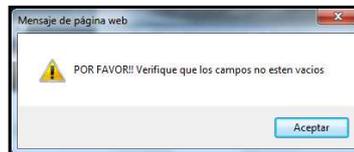
En cuanto a las validaciones que se ha podido realizar tenemos:

- En el campo donde se ingresa el número de cédula solo se podrá escribir números que y serán como máximo 10.

- Si al momento de autenticarse el usuario no ha seleccionado el distrito se le mostrará un mensaje en el que le indica que no ha sido seleccionado y deberá seleccionar la adecuada.



- Si el usuario no ingresa ninguno de los datos la aplicación le arrojará un error y deberá llenar todos los campos requeridos de manera correcta.



- En el desempeño laboral, si uno de los datos como son el año o el semestre no ha sido seleccionado correctamente o no existe le arroja automáticamente un error indicando la razón, de la misma forma le indica al usuario si este espacio está en blanco.
- Para el caso de las estadísticas donde se debe seleccionar la fecha, la aplicación está controlando que el año ingresado no sea mayor que el año actual y lo mismo para el mes.

5.7.Manual de Configuración de la Aplicación Web “SISREHU”

Ver en Anexo 1

5.8.Manual de Usuario de la Aplicación Web “SISREHU”

Ver Anexo 2

CONCLUSIONES

- GWT, un conjunto de herramientas de Google para el desarrollo Web, donde se busca encapsular todos los elementos necesarios para que el desarrollo de aplicaciones Web no sea una tarea traumática y pueda compararse en términos de herramientas, a los desarrollos tradicionales.
- La arquitectura, GWT permite la creación de aplicaciones Web de última generación gracias a una abstracción total del desarrollo del lenguaje de programación Java. Y gracias a sus ventajas el control a particularidades de los navegadores, la optimización de código, las técnicas tradicionales de seguimiento al código, los casos de prueba, la ingeniería de software, y demás elementos de los ciclos de desarrollo particulares, pueden ser implementados con toda naturalidad y fluidez.
- Después de haber comparado y analizado las diferentes características, ventajas y desventajas que presentan los IDE's como Eclipse, Netbeans e IntelliJ, se ha elegido Eclipse como el IDE más adecuado ya que alcanzo el 91.22% el puntaje mas alto, ya que en él se pueden agregar plugins a medida que se va desarrollando logrando así una aplicación escalable y más liviana que es lo ideal al momento de subirla al servidor web y navegar en ella.

- Para la elaboración de la propuesta metodológica para determinar las ventajas y desventajas de la plataforma RIA-GWT se ha utilizado la herramienta analítica FODA la misma que facilita la manera de sistematizar la información que posee dicha plataforma en el mercado reuniendo información de factores externos e internos como son la Fortalezas, Oportunidades, Debilidades y Amenazas., con fin de definir su capacidad competitiva en un período determinado.
- Luego de haber realizado todos los cálculos correspondientes para la demostración de la hipótesis analizando proceso por proceso se puede concluir que efectivamente con el desarrollo de la aplicación web usando la plataforma RIA-GWT para el departamento de recursos humanos de la Refinería Estatal de Esmeraldas se ha podido reducir el tiempo en el flujo de trabajo en un 98.91% en lo concerniente a los procesos que realiza día a día los empleados de dicha entidad.
- Las pruebas y las validaciones que se hacen luego de haber concluido una aplicación web son de mucha importancia para el cumplimiento de los objetivos planteados, y en el caso particular del presente trabajo investigativo no fue la excepción pues, luego de la corrección de posibles errores se logró obtener una aplicación eficaz que realiza los requerimientos planteados.
- En lo relacionado a los aspectos legales, GWT se distribuye con código fuente y con licencia Apache 6.0 (salvo el compilador y la consola de ejecución, de la cual sólo se distribuyen los binarios).

- El desarrollo Web constituye uno de los grandes desafíos del mundo informático en la actualidad. Con la explosión de la Internet se han desarrollado tecnologías impresionantes para hacer sitios Web cada vez más dinámicos, con interfaces de usuario similares a las de escritorio y con niveles de respuesta óptimos donde el tiempo de los usuarios empieza a ser importante y donde cada vez es menos común el desarrollo de sitios donde sea necesario esperar para recibir una respuesta.
- Uno de los grandes adelantos tecnológicos, sin duda es lo que hoy se conoce con el término AJAX. Este conjunto de tecnologías han posibilitado una expansión de las aplicaciones Web respaldadas e impulsadas por grandes informáticos como Google, y gracias a esta tendencia se ha venido desarrollando fuertemente en la Web 2.0.

RECOMENDACIONES

- El compilador especial que incluye GWT no soporta caracteres especiales (vocales acentuadas, etc.) a menos que los ficheros fuente se encuentren codificados usando UTF-8. Se recomienda por tanto usar editores que soporten este tipo de codificación. Esta limitación es sólo para los ficheros compilados por el compilador GWT, es decir, los del lado cliente.
- Después de haber desarrollado la aplicación web y utilizada la base de datos de REE, se puede decir que para un trabajo como este, se debe construir una nueva base de datos ya que no se sabe si la empresa donde se esté realizando la tesis vaya a tener o no reestructuraciones.
- Al momento de plantear los objetivos en la tesis se debe tener en cuenta el tiempo que va a tomar en desarrollar cada uno de estos, para así no tener ningún inconveniente en el transcurso de la tesis.
- Antes de empezar a desarrollar una aplicación con RIA GWT o con cualquier otra herramienta se debe estudiar detenidamente todo lo referente al tema como en este caso programar con JAVA, saber cómo funciona JavaScript y HTML porque aunque no se lo utiliza directamente si se lo hace dentro de GWT; para luego no perder tiempo al momento de realizar la aplicación.

- Al momento de utilizar Eclipse y AS400 se debe saber cómo realizar las conexiones para extraer la información de la base de datos, además qué plugins necesita agregar al proyecto para que este funcione correctamente.

RESÚMEN

Se estudió y analizó la plataforma RIA GWT para el desarrollo de la aplicación web en el departamento de Recursos Humanos de la Refinería Estatal de Esmeraldas.

Para el desarrollo de la investigación se realizó el estudio teórico de conceptos para entender el funcionamiento del framework GWT, se estudio todo sobre el lenguaje de desarrollo JAVA como: definición y característica, lo referente a la máquina virtual de java (JVM), el JRE, entre otros aspectos. Además utilizando el IDE Eclipse y la plataforma RIA GWT se desarrolló e implantó la aplicación web con el objetivo de ayudar a los empleados de la Refinería Estatal de Esmeraldas a obtener información de manera confidencial.

Para la demostración de la hipótesis se empleó el método de entrevistas a los empleados de Recursos Humanos, analizando proceso por proceso, de lo cual se puede concluir que efectivamente con el desarrollo de la aplicación web usando la plataforma RIA-GWT, se ha podido reducir el tiempo en el flujo de trabajo en un 98.91% en lo concerniente a los procesos que realizan los empleados de dicha entidad.

Comparado y analizado los IDE's como Eclipse, Netbeans e IntelliJ, se eligió Eclipse como el más adecuado porque alcanzó el mayor puntaje en lo relacionado al desarrollo de la aplicación web usando la plataforma RIA GWT.

La plataforma RIA-GWT, es de gran ayuda para los desarrolladores de JAVA, porque le permite programar solo en este lenguaje sin preocuparse del código HTML y JavaScript reduciendo con ello tiempo en el desarrollo de la aplicación.

Se recomienda usar el IDE Eclipse como entorno de desarrollo para aplicaciones web basada en la plataforma RIA-GWT.

SUMMARY

Studied and analyzed the GWT RIA platform for web application development in the Human Resources department of the State of Esmeraldas refinery.

For the development of the research was conducted the theoretical study of concepts for understanding the functioning of the GWT framework, they study everything about the Java development language as the definition and characteristics, as regards the Java virtual machine (JVM), Java , among others. Besides using the Eclipse IDE and GWT RIA platform was developed and implemented the web application in order to help employees of the State of Esmeraldas Refinery information confidential.

For the proof of the hypothesis is used the method of interviews with employees of Human Resources, analyzing process by process, which can effectively conclude that the web application development using the RIA-GWT platform, has been reduced time in the workflow by 98.91% with regard to the processes performed by employees of the entity. Compared and analyzed the IDE's like Eclipse, Netbeans and IntelliJ, Eclipse was chosen as the most suitable because it reached the highest score in relation to web application development using GWT RIA platform.

RIA-GWT platform, it is helpful for Java developers because it allows you to program only in this language without worrying about HTML and JavaScript code thereby reducing the development time of the application.

We recommend using the Eclipse IDE as a development environment for Web applications based on GWT RIA platform.

ANEXOS

GLOSARIO

ActionScript

Lenguaje de programación orientado a objetos (OOP), utilizado en especial en aplicaciones web animadas realizadas en el entorno Adobe Flash.

Lenguaje de programación

Es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Programación Orientada a Objeto

La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento.

Adobe flash

Adobe Flash Es una aplicación en forma de estudio que trabaja sobre "fotogramas", destinado a la producción y entrega de contenido interactivo para las diferentes audiencias alrededor del mundo sin importar la plataforma.

CSS

Hojas de estilo en cascada, son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.

Eficiencia

Uso racional de los medios con que se cuenta para alcanzar un objetivo predeterminado.

Etiqueta (o tag)

Marca con tipo que delimita una región en los lenguajes basados en XML.

Fotograma

Elemento en el que se puede definir un cambio en las propiedades de un objeto de una animación.

Fuentes RSS activas

Archivo generado de forma automática por el programa de gestión del weblog que, separando el contenido del diseño.

Gadgets

Dispositivo que tiene un propósito y una función específica, generalmente de pequeñas proporciones, práctico y a la vez novedoso.

Interfaz de usuario

Medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

Lenguaje Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

JavaScript

Lenguaje De Programación Interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Patrón de diseño

Base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Software

Se conoce como software al equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la

realización de tareas específicas, en contraposición a los componentes físicos del sistema, llamados hardware.

Hardware

Corresponde a todas las partes físicas y tangibles de una computadora.

XAML (eXtensible Application Markup Language, Lenguaje Extensible de Formato para Aplicaciones) lenguaje de formato para la interfaz de usuario para la Base de Presentación de Windows (WPF por sus siglas en inglés) y Silverlight(wpf/e), el cual es uno de los "pilares" de la interfaz de programación de aplicaciones .NET en su versión 3.0.

XHTML

Acrónimo inglés de eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.

XML

Extensible Markup Language, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

Página Web

También conocida como una página de **Internet**, es un documento electrónico adaptado para la Web, pero normalmente forma parte de un sitio Web. Su principal característica son los hipervínculos de una página, siendo esto el fundamento de la Web.

Sitio Web

Es un conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.

Aplicación Web

En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Ingeniería de Software

Es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

Servidor Web

Un servidor web es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

El programa implementa el *protocolo HTTP (HyperText Transfer Protocol)* que pertenece a la capa de aplicación del modelo OSI.

Protocolo

Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red. Un protocolo es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales.

HTTP

Es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores.

Navegador Web

Un navegador o navegador web (del inglés, *web browser*) es un programa que permite ver la información que contiene una página web (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local).

Aplicación

Programa informático que permite a un usuario utilizar una computadora con un fin específico. Las aplicaciones son parte del software de una computadora, y suelen ejecutarse sobre el sistema operativo.

BILIOGRAFÍA

- **Aplicaciones Web 100% JAVA con Google Web Toolkit** [en línea]
http://www.finderit.com/index.php?option=com_lyftenbloggie&view=entry&id=53%3AAplicaciones+web+100%25+Java+con+Google+Web+Toolkit&Itemid=143&lang=es
[2010/01/15]
- **Bloque de Conocimiento de Google Web Toolkit (GWT)** [en línea]
<http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=GWT>
[2009/10/10]
- **Caja de Herramientas del Web de Google** [en línea]
http://www.worldlingo.com/ma/enwiki/es/Google_Web_Toolkit#Features
[2009/11/05]
- **Campania Google, Google Web Toolkit** [en línea]
<http://code.google.com/intl/es-ES/webtoolkit/overview.html>
[2009/11/25]
- **Compiladores e IDE´s** [en línea]
<http://casidiablo.net/compiladores-e-ide%E2%80%99s/>
[2010/01/013]
- **ECLIPSE (Software)** [en línea]
http://es.wikipedia.org/wiki/Eclipse_%28software%29
[2010/01/10]

- **Google Web Toolkit** [en línea]
http://www.slideshare.net/Guido_De_Benedetti/Google-Web-Toolkit-904220
[2009/11/24]
- **Google Web Toolkit, GWT Java Ajax Programming** [en línea]
https://www.packtpub.com/sites/default/files/sample_chapters/Google-Web-Toolkit-Ajax-Creating-a-New-GWT-Application-Sample-Chapter.pdf
[2009/10/10]
- **Google Web Toolkit (GWT), Progress Software y Servicios 2010** [en línea]
<http://www.slideshare.net/pgdesarrollo/5-introduccion-gwt>
[2009/11/05]
- **Grupo Mediawiki, GWT** [en línea]
<http://www.grupos2p.org/wiki/index.php/GWT>
[2009/11/05]
- **Guía breve sobre Google Web Toolkit** [en línea]
http://webdiis.unizar.es/~jogracia/other/introduccion_a_gwt.htm
[2009/11/24]
- **GWT** [en línea]
<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/GWT>
[2010/01/15]
- **GXT, GWT, Java eclipse Tutorial (Parte III)** [en línea]
<http://dardison.wordpress.com/2009/09/29/gxt-gwt-java-eclipse-tutorial-parte-iii/>
[2010/01/15]

- **Ingeniería Informática Empresarial** [en línea]
<http://s3.amazonaws.com/ppt-download/presentaciongwt-091202060114-phpapp02.pdf?Signature=h6lGbfkxO4oIIIe8TbCwaGa3awo%3D&Expires=1275865680&AWSAccessKeyId=AKIAJLJT267DEGKZDHEQ>
 [2009/10/10]
- **Introduction to GWT Remote Procedure Calls (RPC) (with example application)**
 [en línea]
<http://www.ajaxmatters.com/2006/05/introduction-to-gwt-remote-procedure-calls-rpc-with-example-application/>
 [2010/01/13]
- **JAVA SE 6 Documentation** [en línea]
<http://java.sun.com/javase/6/docs/index.html>
 [2010/01/10]
- **Lenguaje de Programación en Java** [en línea]
http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java
 [2010/01/08]
- **Maquina Virtual Java** [en línea]
http://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java
 [2010/01/08]
- **Tutorial (1.0) de Introducción a Google Web Toolkit** [en línea]
<http://casidiablohost.googlepages.com/Introduccion-al-Google-Web-Toolkit.pdf>
 [2009/10/29]

**ANEXO 1: MANUAL DE CONFIGURACIÓN DE
LA APLICACIÓN WEB “SISREHU”**

Manual de Configuración

“Sitio Web SISREHU”

Documento Desarrollado por los autores Henry Fabricio Murillo Vélez y Mirian Susana Riera Idrovo, como parte de la tesis realizado en la Refinería Estatal de Esmeraldas el sitio web “SISREHU”.

© Copyright 2010 Todos los derechos reservados

1. PRESENTACIÓN

Este manual servirá de guía a aquellas personas que estén involucradas en la instalación y configuración del sitio web. Describe la manera correcta como se debe configurar el servidor, describiendo los pasos necesarios para obtener el archivo .war, el mismo que nos permitirá empaquetar todo el proyecto para ser subido en el servidor de aplicaciones web apache.

Además, se presenta paso a paso todo lo necesario en cuanto a la configuración del servidor apache, así como también los requisitos necesarios para cumplir con éxito todo el proceso y tener como resultado una correcta operatividad del Sitio Web “SISREHU”.

2. INTRODUCCIÓN

Sitio Web “SISREHU” es una aplicación web diseñada para el personal de Recursos Humanos de la Refinería Estatal de Esmeraldas con el fin de mostrar la información confidencial y necesaria con solo ingresar datos como: el número de cédula de identidad, la clave y el Distrito de Refinación al que pertenece.

3. REQUISITOS MÍNIMOS INDISPENSABLES

Hardware

- PC con procesador Pentium III o superior.
- 512 MB de memoria RAM.
- Disco duro de 40 Gb
- Monitor
- Teclado
- Mouse

Software

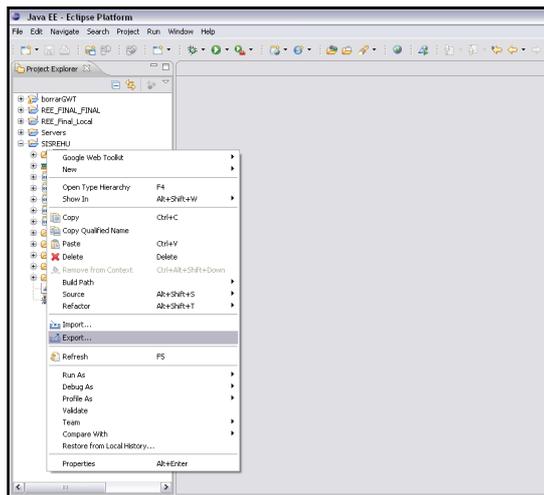
- Sistema Operativo Microsoft Windows XP/Vista /Win7
- Microsoft .Net Framework 2.0 o superior.
- La máquina virtual de Java.

- Servidor de Aplicaciones Web (Apache Tomcat 6.0)
- IDE Eclipse 3.4.
- GWT 1.5

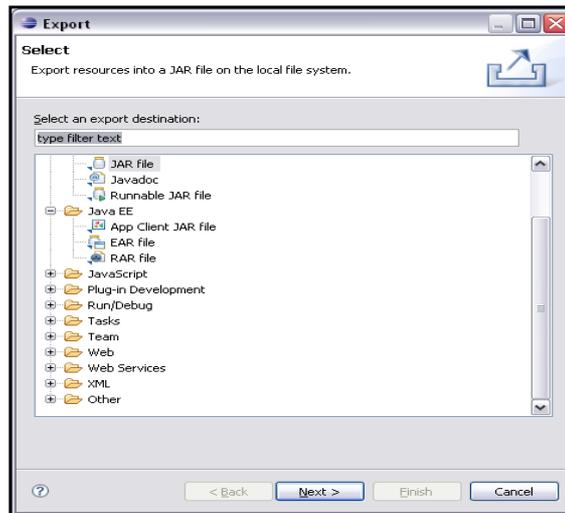
4. PASOS PARA CONFIGURAR Y SUBIR LA APLICACIÓN WEB

Pasos que se deben hacer dentro de eclipse:

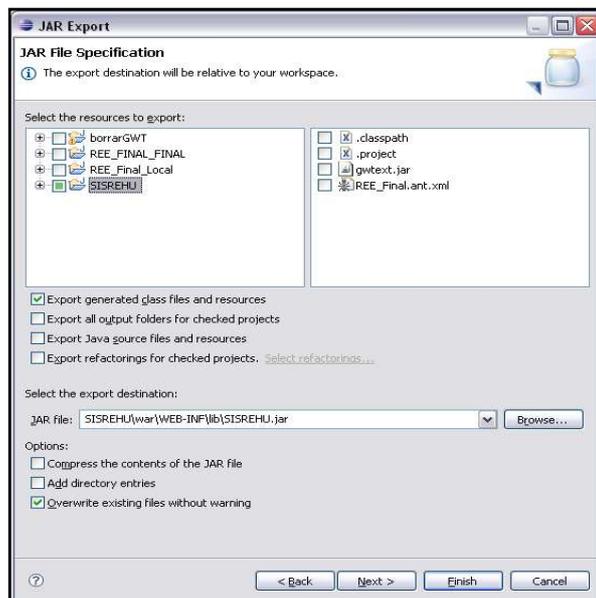
1. Creamos una carpeta dentro del proyecto SISREHU llamada **war**, y dentro de esta otra llamada **WEB-INF** y dentro de ésta otra carpeta llamada **lib**.
2. Una vez hecho esto, nos vamos al eclipse y le damos clic derecho sobre el proyecto y escogemos la opción **Export** como se muestra en la figura.



3. Creamos un archivo con la extensión **.jar** seleccionando JAR file como se muestra en la figura, y le damos clic en **next**, como se muestra a continuación.

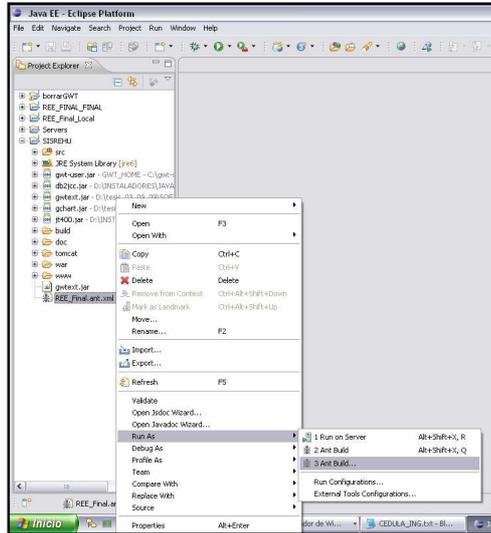


4. Se marca solo la carpeta SRC del proyecto SISREHU, como se muestra en la figura, luego damos clic en el botón Browser y buscamos la ruta en donde se encuentra nuestra carpeta lib, que se encuentra dentro de WEB-INF, como se muestra, y damos clic en **Finish**.

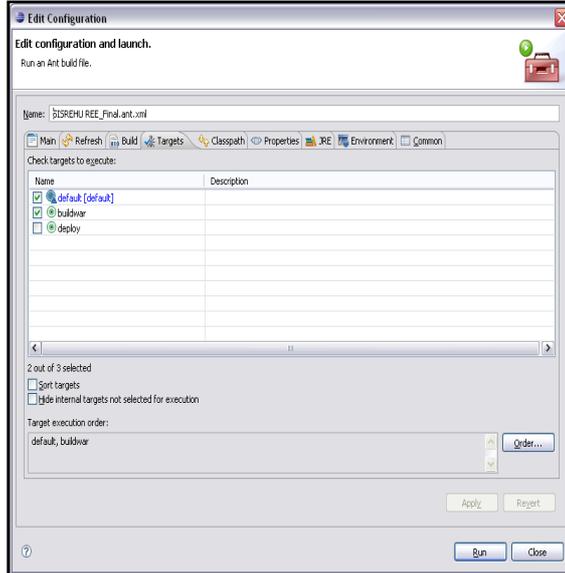


5. Luego de esto le presionamos la tecla F5 para actualizar lo que se ha realizado.

6. Luego sobre el archivo *SISREHU.ant.xml*, le damos clic derecho escogemos la opción Run As, Ant Build y le damos clic, como se muestra a continuación.



7. Luego de ello, seleccionamos las dos primeras opciones como se muestra en la figura, y damos clic en *Apply* (si no se encuentra habilitado), y en *Run*.



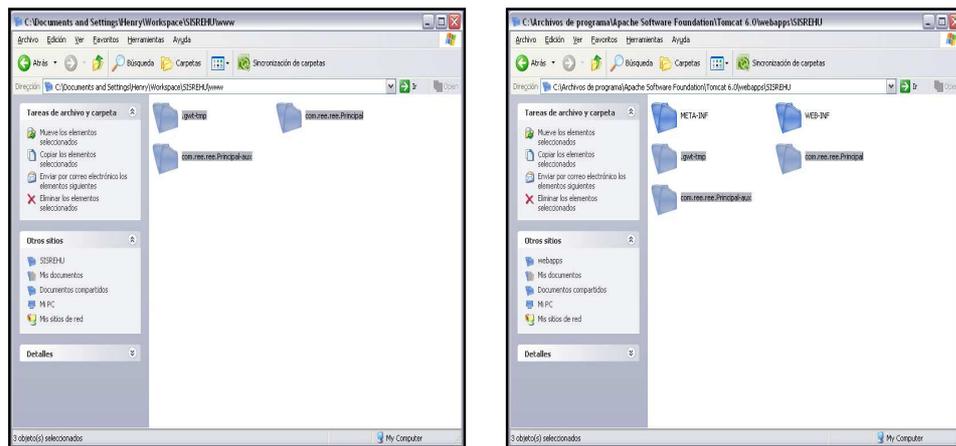
8. Luego actualizamos todo el proyecto para que se guarden todos los cambios.

Pasos que se deben hacer dentro del Servidor de Aplicaciones Tomcat:

1. Nos situamos dentro del servidor de aplicaciones tomcat, para ello nos vamos a C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0\webapps, y creamos una carpeta llamada SISREHU.

Sobre la carpeta creada descomprimos el archivo SISREHU.ant.xml, que se generó dentro del proyecto los pasos anteriores (dentro de eclipse).

2. Luego nos situamos en la carpeta *www* que se encuentra en la carpeta del proyecto, copiamos todos los archivos que se encuentran ahí, y los pegamos dentro de la carpeta SISREHU que se encuentra dentro del tomcat como se muestra a continuación.



3. Dentro del apache, sacamos todos los archivos que se encuentra en la carpeta donde se alojan todas las clases del cliente en este caso la carpeta *com.ree.ree.Principal* y la pegamos dentro de la carpeta SISREHU que se encuentra en el servidor.
4. Dentro del apache mismo nos situamos en la carpeta WEB-INF/lib sacamos la librería .SISREHU.jar y la descomprimos en el mismo lugar donde descomprimos el paso anterior.
5. Finalmente abrimos el explorador, ingresamos la dirección donde se encuentra alojada nuestra aplicación, en empezamos a navegar según el manual de usuario.

**ANEXO 2: MANUAL DE USUARIO DE LA
APLICACIÓN WEB “SISREHU”**

Manual de Usuario

“Sitio Web SISREHU”

Documento Desarrollado por los autores Henry Fabricio Murillo Vélez y Mirian Susana Riera Idrovo, como parte de la tesis realizado en la Refinería Estatal de Esmeraldas el sitio web “SISREHU”.

© Copyright 2010 Todos los derechos reservados

1. PRESENTACIÓN

Este manual servirá de guía a aquellas personas que estén involucradas en el manejo del sitio web. Describe la funcionalidad y el uso del Sitio Web “SISREHU”, el mismo que admite llevar una serie de tareas secuenciales que permita culminar el proceso con éxito.

Se presenta información detallada de un correcto manejo del Sitio Web “SISREHU”, para lo que se deberá leer secuencialmente sin obviar algún paso. Muestra la manera más eficiente de manejo y de obtener provecho ya que presenta toda la información con el fin de que el usuario pueda tener una completa visión del software antes de iniciar con su funcionamiento.

2. INTRODUCCIÓN

Sitio Web “SISREHU” es una aplicación web diseñada para el personal de Recursos Humanos de la Refinería Estatal de Esmeraldas con el fin de mostrar la información confidencial ingresando los datos como: el número de cédula de identidad, la clave y el distrito al que pertenece.

3. OBJETIVOS SISTEMA

- Automatizar la información matemáticas a las barras estadísticas en el Sitio Web “SISREHU”.
- Reducir el tiempo al personal de Recursos Humanos al momento de consultar sus datos confidenciales.

4. CAPACIDADES DEL SISTEMA.

Este sistema es capaz de proporcionar:

- Ofrecer una interfaz gráfica amigable facilitando el manejo por parte de los usuarios.
- Brindar seguridades en el manejo del sitio web permitiendo el acceso únicamente al personal autorizado.
- Controlar campos vacíos

5. REQUISITOS MÍNIMOS INDISPENSABLES

Hardware

- PC con procesador Pentium III o superior.
- 512 MB de memoria RAM.
- Disco duro de 40 GB
- Monitor
- Teclado
- Mouse

Software

- Sistema Operativo Microsoft Windows XP/2002
- Microsoft .Net Framework 2.0.

6. COMO UTILIZAR EL PROGRAMA

El programa tiene la siguiente estructura:

6.1 Pantalla de Principal

El sitio web presenta la pantalla principal con los datos generales (Inicio, Misión/Visión, Quiénes somos, Contáctenos, Acerca de...) y las noticias.





6.2 Autenticación del Personal

Llenar los campos vacíos del Inicio de Sesión como es: el número de cédula, la clave personal y seleccionar el distrito al que pertenece y luego dar click en **Iniciar Sesión** para las consultas correspondientes.

This screenshot shows the login form on the EP Petroecuador website. The form is titled 'Inicio Sesión' and includes a green person icon. It contains three input fields: 'Cédula:' with the value '0703103556', 'Contraseña:' with a masked password '*****', and 'Distrito:' with a dropdown menu showing 'Esmeraldas'. Below the fields is an 'Aceptar' button.

Nota: La cedula de identidad se ingresa sin el guion, es muy importante que se llenen los datos del empleado.

A continuación se muestra la página donde se autentico el empleado y se habilita el submenú de Consultas, cuando es un usuario normal en el submenú no se mostrará la opción de Estadísticas ya que esa opción es solo para el usuario Administrador.



6.3 Consultar la Asistencia

Para consultar la asistencia del empleado elegimos la opción **Consultas** del menú principal y luego dar click en **Asistencia** y se mostrará la información.



Además se pueden filtrar los datos por una fecha determinada como se indica en la siguiente figura.

EMPRESA ESTATAL PETROLEOS DEL ECUADOR

EP PETROECUADOR

DISTRITO: REFINERIA ESTATAL DE ESMERALDAS

Inicio Misión Visión Quienes Somos Contáctenos Consultas Acerca de

Asistencia [Cerrar Sesión](#)

N° Empleado: 32167 Cédula: 0703103556

Nombres: BAUTISTA YEPEZ LORENA LILLYBETH

Ingrese la fecha que desea buscar:

Ingrese: Año / Mes / Día: ejemplo: 20060101

HECHA	MAÑANA		TARDE	
	Hora Entrada	Hora Salida	Hora Entrada	Hora Salida
20060606	07:29	11:53	13:34	16:51

NOTICIAS

PETROINDUSTRIAL firma contratos de trabajo con personas con necesidades especiales.- Con el objetivo de completar el porcentaje del tres por ciento del total de trabajadores que laboran en la filial de Petroecuador... [Ver más...](#)

Cuestionan a Petromineral.- La adjudicación de la Estación de Seroya para la extracción de gas ha enfrentado, con o sin intencionalidades, a dos de las empresas participantes... [Ver más...](#)

6.4 Consultar los Cursos Recibidos

Para visualizar los cursos recibidos del empleado seleccionar la opción **Consultas** del menú principal y luego elegir **Cursos Recibidos**, se desplegará los cursos recibidos.

Consultas **Acerca de**

- Asistencia
- Cursos Recibidos**
- Datos Familiares
- Datos Personales
- Desempeño Laboral
- Estadísticas
- Historial Financiero
- Permisos
- Record Académico
- Saldo Vacacional

EMPRESA ESTATAL PETROLEOS DEL ECUADOR

EP PETROECUADOR

DISTRITO: REFINERIA ESTATAL DE ESMERALDAS

Inicio Misión Visión Quienes Somos Contáctenos **Consultas** Acerca de

Cursos Recibidos [Cerrar Sesión](#)

N° Empleado: 30922 Cédula: 0800666836

Nombres: CARRERA PEREA DANA

No.	Evento	Duración	País	Año/mes
01	SISTEMA OPERATIVO MS-DOS	35	ECUADOR	9409
02	WORD PERFECT	20	ECUADOR	9409

NOTICIAS

PETROINDUSTRIAL firma contratos de trabajo con personas con necesidades especiales.- Con el objetivo de completar el porcentaje del tres por ciento del total de trabajadores que laboran en la filial de Petroecuador... [Ver más...](#)

6.5 Consultar los Datos Familiares

Para consultar los datos familiares del empleado se debe seleccionar en el menú principal **Consultas** luego dar click en **Datos Familiares** y se visualizará los datos familiares del empleado.



A screenshot of the 'Datos Familiares' page in the EP Petroecuador system. The page header includes the company logo and name 'EMPRESA ESTATAL PETROLEOS DEL ECUADOR' and 'EP PETROECUADOR'. Below the header is a navigation bar with 'Inicio', 'Misión Visión', 'Quiénes Somos', 'Contáctenos', 'Consultas', and 'Acerca de'. The main content area is titled 'Datos Familiares' and shows the employee's ID (Nº Empleado: 32167) and ID card number (Cédula: 0703103556). The employee's name is 'BAUTISTA YEPEZ LORENA LILLYBETH'. A table lists family members:

Num.	Nombre del familiar	Parentesco	Fecha de nacimiento
01	RODRIGUEZ PORTES DAVID LEONARDO	1 CONJUGE	19771231
02	RODRIGUEZ BAUTISTA JUREN DAVID	2 HIJO(A)	20050610
03	RODRIGUEZ BAUTISTA JOQUE LEONARDO	2 HIJO(A)	20100204

On the right side, there is a 'NOTICIAS' section with a link to 'Cerrar Sesión' and a news item about Petroindustrial.

6.6 Consultar los Datos Personales

Para consultar los datos personales del empleado se debe seleccionar en el menú principal **Consultas** luego dar click en **Datos Personales** y se mostrará la información personal del empleado.



A screenshot of the 'Datos Personales' page in the EP Petroecuador system. The page header includes the company logo and name 'EMPRESA ESTATAL PETROLEOS DEL ECUADOR' and 'EP PETROECUADOR'. Below the header is a navigation bar with 'Inicio', 'Misión Visión', 'Quiénes Somos', 'Contáctenos', 'Consultas', and 'Acerca de'. The main content area is titled 'Datos Personales' and shows the employee's ID (Nº Empleado: 32167) and ID card number (Cédula: 0703103556). The employee's name is 'BAUTISTA YEPEZ LORENA LILLYBETH'. A table lists personal data:

Fecha de Nacimiento / Edad	19790721 31 AÑOS
Relación Laboral	4 EVENTUAL EMPRESA
Cargo	ANALISTA DE APLICACIONES
Código Dependencia	7050001000000
Nivel / Sueldo Básico	10 1160.05 Dólares
Estado	A ACTIVO
Tiempo Petroecuador	4 AÑOS 6 MESES
Tiempo otras Instituciones	0 AÑOS 0 MESES
TOTAL	4 AÑOS 6 MESES

On the right side, there is a 'NOTICIAS' section with a link to 'Cerrar Sesión' and a news item about Petroindustrial.

6.7 Consultar el Desempeño Laboral

Para consultar el desempeño laboral del empleado elegimos del menú principal la opción de **Consultas**, click en **Desempeño Laboral** luego se debe seleccionar el periodo y el año para desplegar la información correspondiente al periodo y año seleccionado.

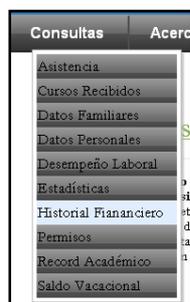


The page shows the 'Historial Desempeño Laboral' for employee RHOR CEVALLOS RENE FEDERICO. It includes fields for 'Evaluado' (N° Empleado: 31979, Cédula: 0802491183) and 'Evaluador' (N° Empleado: 41549, Cédula: 1708503287). A table lists evaluation factors and their degrees.

FACTORES	GRADO
Iniciativa	4
Disciplina	2
Comp. Trabajo	3
Interés y Participación	0
Adaptación	0
TOTAL	23
PROMEDIO	2.0833333333333333

6.8 Consultar el Historial Financiero

Para consultar el historial financiero del empleado se debe seleccionar en el menú principal **Consultas** y después dar click en **Historial Financiero** y se desplegará la información correspondiente.



The page shows the 'Historial Financiero' for employee BAUTISTA YEPEZ LORENA LILLYBETH. It includes fields for 'N° Empleado: 32167' and 'Cédula: 0703103556'. A table lists monthly and additional payments from 2006 to 2007.

Fecha	PAGOS MENSUALES			PAGOS ADICIONALES			
	Ingresos	Desc.	Neto	Fondo Reserva	Dec Tercero	Dec Cuarto	Inc. Jefatura
2006-11	946.72	254.46	712.26	0.0	0.0	0.0	0.0
2006-12	1171.7	240.19	930.87	0.0	441.16	0.0	0.0
2007-01	925.29	447.83	475.55	0.0	0.0	0.0	0.0
2007-02	916.16	612.40	303.76	0.0	0.0	0.0	0.0
2007-03	1035.43	707.53	327.90	0.0	0.0	0.0	0.0

Además se pueden filtrar los datos por una fecha determinada como se indica en la siguiente figura.

6.9 Consultar el Record Académico

Para consultar el record académico del empleado seleccionar en el menú principal la opción **Consultas** y a continuación click en **Record Académico** y se mostrará la información.

No.	Título	Duración	Año	Estado	Ciudad
		Año Mes	Obt.		
01	TECNOLÓGICO EN COMPUTACION	3	2003	T	ESMERALDAS
02	INGENIERIA EN SISTEMAS Y COMPUTACION	6	2008	T	ESMERALDAS
03	BACHILLER EN INFORMÁTICA	0	1997	T	ESMERALDAS

6.10 Consultar el Saldo Vacacional

Para mostrar el saldo vacacional dar click en **Consultas** del menú principal y luego seleccionar **Saldo Vacacional**.



SOLO PARA LOS USUARIOS ADMINISTRADORES DE LA REE

6.11 Consultar las Estadísticas

Para visualizar las estadísticas elegir la opción **Consultas** que se encuentra en el menú principal y luego dar click en **Estadísticas** donde se mostrará un submenú que se detallará posteriormente.



Mostrar el Desempeño Laboral.- para visualizar en barras estadísticas el desempeño laboral de todos los empleados de Recursos Humanos de REE, seleccionar el periodo y año que quiera ver los resultados de los empleados.



Mostrar el número de Hombres y Mujeres.- aquí se desplegará en barras estadísticas el total de hombres y mujeres que existen en el área de Recursos Humanos de REE.



Mostrar el número de empleados con Nivel Académico.- se visualiza en barras estadísticas el número de empleados con los diferentes niveles académicos como: primaria, secundaria y universitaria.



Mostrar el número profesionales principales.- aquí se desplegará en barras estadísticas el total de empleados en las diferentes profesiones tomadas en cuenta como las principales o más pedidas por los Usuario como: Ing. Químico, Ing. Mecánico, Ing. Sistemas, etc.



Mostrar el número de trabajadores contratados en un determinado mes.- aquí se mostrará en barras estadísticas el total de empleados contratados en un mes determinado y a qué área de trabajo ingreso, para esto se debe seleccionar una fecha específica en el calendario para visualizar la información.



Mostrar el número de trabajadores en las diferentes áreas.- se visualizará mediante barras estadísticas el total de empleados que existen en las diferentes áreas de trabajo de los empleados de Recursos Humanos de REE.

