



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

**ESTUDIO COMPARATIVO DE LAS METODOLOGÍAS UWE Y OOWS PARA
MEJORAR LA PRODUCTIVIDAD EN EL DESARROLLO DE APLICACIONES
WEB. CASO PRÁCTICO: CENTRO PARVULARIO POLITECNICO**

TESIS DE GRADO

PREVIA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN SISTEMAS INFORMÁTICOS

Presentado por:

JENNY PAOLA CORO ADRIANO

RIOBAMBA-ECUADOR

2014

Agradezco a Dios y a mis padres por brindarme sus bendiciones y apoyo incondicional, a mis hermanos y amigos por estar conmigo cuando los necesité, a mi hija y mi esposo por su paciencia y cariño, a la ESPOCH institución que me abrió sus puertas y, mis queridos maestros quienes permitieron mi formación académica, de manera especial al director y miembro de tesis, que con sus conocimientos fueron una guía en el desarrollo de la misma; pues gracias a todos ellos he luchado constantemente para cumplir mis metas, y hacer posible este sueño.

Jenny Paola Coro Adriano

La realización de este trabajo esta dedicada con mucho amor a mi hija Domenica Samantha y mi esposo Luis Marcelo, que con su paciencia y comprensión permitieron cumplir este sueño, a mis padres Teresa y Manuel que estuvieron junto a mí, brindando su apoyo y fortaleza día a día, y a toda mi familia y amigos con quienes compartí dificultades, éxitos y diferentes anécdotas durante el trayecto .

Jenny Paola Coro Adriano

FIRMAS RESPONSABLES Y NOTAS

NOMBRES	FIRMA	FECHA
ING. GONZALO SAMANIEGO E. DECANO DE LA FACULTAD DE INFORMATICA Y ELECTRONICA	_____	_____
DR. JULIO SANTILLAN CASTILLO DIRECTOR DE LA ESCUELA DE INGENIERIA EN SISTEMAS	_____	_____
DR. JULIO SANTILLAN CASTILLO DIRECTOR DE TESIS	_____	_____
ING. FERNANDO MEJIA MIEMBRO DE TESIS	_____	_____
DIRECTOR DEL CENTRO DE DOCUMENTACIÓN	_____	_____

NOTA: _____

RESPONSABILIDAD DEL AUTOR

Yo, Jenny Paola Coro Adriano, soy responsable 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.

Jenny Paola Coro Adriano

ÍNDICE DE ABREVIATURAS

COCOMO: Modelo Constructivo de Costos (COConstructive COst MOdel)

DBMS: Sistema de Administración de Base de Datos

EORM: Metodología de Relación entre Objeto

HDM: Modelo de Diseño de Hipertexto

HFPM: Hypermedia Flexible Process Modeling,

IWEB: Ingeniería Web

JSF: Java Server Faces

JEE: Java Enterprise Edition.

MVC: Modelo de vista controlador

NDT: Técnicas de Desarrollo Navegacional

OOHDM: Método de Diseño de Hipermedia Orientado a Objetos

OOWS: Solución Web Orientado a Objetos

RNA: Relationship-Navegational Analysis

SIW: Sistemas de información web

SQL: Structured Query Language, lenguaje de acceso a bases de datos

UML: Lenguaje de Modelamiento Unificado

UWE: UML basado en Ingeniería Web

WSDM: Método de Diseño de Sitios Web

ÍNDICE GENERAL

CONTENIDO

ÍNDICE DE ABREVIATURAS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCION

CAPÍTULO I

MARCO REFERENCIAL

1.1.	ANTECEDENTES.....	- 17 -
1.2.	JUSTIFICACIÓN DEL TEMA	- 19 -
1.2.1.	JUSTIFICACIÓN TEÓRICA	- 19 -
1.2.2.	JUSTIFICACIÓN METODOLÓGICA	- 20 -
1.2.3.	JUSTIFICACIÓN PRÁCTICA.....	- 20 -
1.3.	OBJETIVOS	- 21 -
1.3.1.	OBJETIVO GENERAL	- 21 -
1.3.2.	OBJETIVOS ESPECÍFICOS.....	- 21 -
1.4.	RECURSOS	- 21 -

CAPÍTULO II

MARCO TEÓRICO

2.1.	INGENIERÍA WEB.....	- 23 -
2.2.	INGENIERÍA DE REQUISITOS EN LA INGENIERÍA WEB.....	- 25 -
2.3.	APLICACIÓN WEB.....	- 27 -
2.3.1.	CATEGORÍA DE APLICACIONES WEB.....	- 27 -
2.4.	DEFINICIÓN DE METODOLOGÍA WEB	- 28 -
2.4.1.	EVOLUCIÓN DE LAS METODOLOGÍAS DE DESARROLLO WEB.	- 29 -
2.4.2.	TIPOS DE METODOLOGÍAS WEB.....	- 29 -
2.4.3.	VENTAJAS DE UTILIZAR UNA METODOLOGIA.....	- 32 -
2.5.	UWE: UML BASED WEB ENGINEERING.....	- 33 -
2.5.1.	PROCESO O FASES.....	- 33 -
2.5.2.	VENTAJAS Y DESVENTAJAS	- 37 -
2.6.	OOWS: OBJECT ORIENTED WEB SOLUTION	- 38 -
2.6.1.	PROCESO.....	- 39 -
2.6.1.1.	Especificación del Sistema	- 39 -
2.6.1.2.	Desarrollo de la Solución.	- 41 -

2.6.2.	CONSTRUCCION DEL MODELO NAVEGACIONAL	- 42 -
2.6.3.	CONSTRUCCION DEL MODELO DE PRESENTACIÓN.....	- 43 -
2.6.4.	VENTAJAS Y DESVENTAJAS	- 45 -
2.7.	PRODUCTIVIDAD	- 45 -
2.8.	JSF (JAVA SERVER FACES)	- 48 -
2.8.1.	ARQUITECTURA JAVA	- 48 -

CAPÍTULO III

MARCO METODOLÓGICO

3.1.	DISEÑO DE LA INVESTIGACIÓN	- 50 -
3.2.	TIPO DE ESTUDIO	- 51 -
3.3.	HIPOTESIS.....	- 51 -
3.4.	DETERMINACIÓN DE INDICADORES Y METRICAS	- 51 -
3.4.1.	INDICADORES.....	- 53 -
3.4.2.	MÉTRICAS	- 55 -
3.5.	MÉTODO DE COMPARACIÓN DE METODOLOGÍAS.....	- 56 -
3.6.	TÉCNICAS Y HERRAMIENTAS UTILIZADOS PARA LA MEDICIÓN	- 57 -
3.6.1.	TECNICA DE MEDICIÓN	- 58 -
3.6.2.	HERRAMIENTA DE MEDICIÓN	- 59 -
3.7.	TÉCNICAS Y FUENTES DE RECOPIACIÓN DE INFORMACIÓN.....	- 59 -
3.8.	PROCESAMIENTO DE LA INFORMACIÓN	- 60 -
3.8.1.	CONSTRUCCIÓN DE PROTOTIPOS	- 61 -
3.8.2.	REALIZACIÓN DE LA COMPARACIÓN	- 61 -
3.8.3.	OBTENCIÓN DE RESULTADOS	- 61 -
3.8.4.	ANÁLISIS DE RESULTADOS	- 62 -

CAPÍTULO IV

ANÁLISIS COMPARATIVO DE LAS METODOLOGÍAS WEB UWE Y OOWS

4.1.	DESARROLLO DE LOS PROTOTIPOS.....	- 63 -
4.1.1.	PROTOTIPO DE LA METODOLOGÍA UWE.....	- 63 -
4.1.1.1.	Fase de Análisis de Requisitos	- 63 -
4.1.1.2.	Fase de Diseño Conceptual.....	- 65 -
4.1.1.3.	Fase de Diseño Navegacional.....	- 67 -
4.1.1.4.	Fase de Presentación.....	- 69 -
4.1.1.5.	Modelos complementarios.....	- 70 -
4.1.1.6.	Base de Datos	- 71 -
4.1.1.7.	Implementación o programación de la aplicación	- 72 -
4.1.2.	PROTOTIPO DE LA METODOLOGÍA OOWS	- 73 -
4.1.2.1.	Especificación del Sistema	- 74 -

4.1.2.2. Desarrollo de la Solucion	- 80 -
4.2. EJECUCIÓN DE LA EVALUACIÓN.	- 81 -
4.3. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....	- 92 -
4.4. COMPROBACIÓN DE LA HIPÓTESIS DE INVESTIGACIÓN.....	- 95 -
CAPÍTULO V	
APLICACIÓN WEB PARA EL PARVULARIO POLITÉCNICO	
5.1. GESTIÓN DEL PROYECTO.....	- 100 -
5.1.1. ESTRUCTURA ORGANIZACIONAL.....	- 100 -
5.1.2. FUNCIONALIDAD.....	- 101 -
5.1.3. DESCRIPCIÓN DEL SISTEMA.....	- 102 -
5.1.3.1. Perspectiva del producto.....	- 102 -
5.1.3.2. Diseño de interfaces.....	- 103 -
5.2. DESARROLLO DE LA METODOLOGÍA OOWS	- 105 -
5.2.1. ESPECIFICACIÓN DEL SISTEMA.....	- 106 -
5.2.1.1. Especificación de Requisitos	- 106 -
5.2.1.2. Modelado Conceptual.....	- 118 -
5.2.1.3. Modelado de Navegación	- 130 -
5.2.1.4. Modelo de Presentación.....	- 133 -
5.2.2. DESARROLLO DE LA SOLUCION.....	- 139 -
5.2.2.1. Capa de Presentación.....	- 139 -
5.2.2.2. Capa de Aplicación.....	- 144 -
5.2.2.3. Capa de Base de Datos	- 146 -
CONCLUSIONES	
RECOMENDACIONES	
RESUMEN	
SUMMARY	
GLOSARIO	
ANEXOS	
BIBLIOGRAFIA	

ÍNDICE DE FIGURAS

Figura II. 1: Clasificación de Aplicaciones.....	- 28 -
Figura II. 2: Métodos de Ingeniería Web	- 30 -
Figura II. 3: Paquetes del Nivel alto.....	- 34 -
Figura II. 4: Modelado Conceptual	- 35 -
Figura II. 5: Modelo de Presentación UWE.....	- 36 -
Figura II. 6: Arquitectura Funcional de la Metodología OOWS.....	- 39 -
Figura II. 7: Mapa Navegacional	- 42 -
Figura II. 8: Modelo de Presentación.....	- 43 -
Figura IV. 9: Diagrama de Análisis de Requisitos.....	- 64 -
Figura IV. 10: Diagrama de Usuario de ingreso a modulo	- 64 -
Figura IV. 11: Diagrama de Usuario de ingreso de datos de padres	- 65 -
Figura IV. 12: Diagrama de Usuario de Ingreso de datos de Representante.....	- 65 -
Figura IV. 13: Diagrama de Componentes	- 66 -
Figura IV. 14: Diagrama de Objetos	- 66 -
Figura IV. 15: Refinamiento del Modelo Conceptual.....	- 67 -
Figura IV. 16: Vista de Integración con el modelo de Navegación	- 68 -
Figura IV. 17: Vista Estructural del Modelo Navegacional.....	- 68 -
Figura IV. 18: Vista de Comportamiento.....	- 69 -
Figura IV. 19: Modelo de Presentación	- 69 -
Figura IV. 20: Diagrama del Modelo de Usuarios	- 70 -
Figura IV. 21: Diagrama del Modelo de Adaptación.....	- 70 -
Figura IV. 22: Diagrama de Actividades	- 71 -
Figura IV. 23: Diagrama de Base de Datos.....	- 72 -
Figura IV. 24: Implementación del Prototipo UWE	- 72 -
Figura IV. 25: Pantalla de ingreso al módulo	- 73 -
Figura IV. 26: Pantalla de ingreso de datos	- 73 -
Figura IV. 27: Diagrama de Caso de uso del Prototipo2	- 74 -
Figura IV. 28: Vista estructural del modelo conceptual.....	- 75 -
Figura IV. 29: Diagrama de Transición de Estados	- 76 -
Figura IV. 30: Diagrama de Secuencia	- 76 -
Figura IV. 31: Diagrama de usuario secretaria	- 77 -

Figura IV. 32: Diagrama de Navegación	- 78 -
Figura IV. 33: Contexto de Inscripción.....	- 79 -
Figura IV. 34: Contexto de Matrícula	- 79 -
Figura IV. 35: Contexto de Alumno	- 79 -
Figura IV. 36: Contexto de Representante	- 80 -
Figura IV. 37: Contexto de Padres	- 80 -
Figura IV. 38: Implementación del Prototipo OOWS.....	- 81 -
Figura IV. 39: Gráfica de resultados de Tipos de requerimientos.....	- 82 -
Figura IV. 40: Gráfica del resultado de la Propuesta Funcional	- 84 -
Figura IV. 41: Gráfica del resultado del Tiempo de diseño	- 85 -
Figura IV. 42: Gráfica del resultado de Modelo de producto	- 87 -
Figura IV. 43: Gráfica del resultado de Modelo de proceso	- 88 -
Figura IV. 44: Puntos de función en COCOMO.....	- 89 -
Figura IV. 45: Estimación de productividad de diseño temprano en COCOMO.....	- 89 -
Figura IV. 46: Estimación de productividad en Post Arquitectura	- 90 -
Figura IV. 47: Gráfica de resultados del estudio comparativo.....	- 96 -
Figura IV. 48: Formulas de Media y Desviación Estándar	- 97 -
Figura IV. 49: Formula de Intervalo de Confianza	- 97 -
Figura IV. 50. Intervalo de confianza que aprueba la hipótesis.....	- 99 -
Figura V. 51: Estructura Organizacional	- 101 -
Figura V. 52: Organigrama Institucional	- 101 -
Figura V. 53: Perspectiva del producto.....	- 103 -
Figura V. 54: Diseño de la pantalla de acceso o inicio de sesión	- 104 -
Figura V. 55: Diseño de la pantalla principal	- 104 -
Figura V. 56: Diseño de pantalla de matrícula.....	- 105 -
Figura V. 57: Diagrama de Caso de uso para la captura de requisitos.....	- 107 -
Figura V. 58: Vista estructural del modelo conceptual.....	- 119 -
Figura V. 59: Diagrama de transición de estados.....	- 123 -
Figura V. 60: Diagrama de secuencia del objeto AnioLectivo	- 124 -
Figura V. 61: Diagrama de Secuencia del objeto inscripción	- 124 -
Figura V. 62: Diagrama de secuencia del objeto matrícula	- 125 -
Figura V. 63: Diagrama de secuencia de registro de alumno.....	- 125 -
Figura V. 64: Diagrama de secuencia del objeto registro de pago.....	- 126 -
Figura V. 65: Diagrama de secuencia del objeto Maestro	- 126 -
Figura V. 66: Diagrama de secuencia del objeto materia	- 127 -

Figura V. 67: Diagrama de secuencia del objeto curso.....	- 127 -
Figura V. 68: Diagrama de secuencia del objeto paralelo.....	- 128 -
Figura V. 69: Diagrama de secuencia del objeto cuentas de materia asignada.....	- 128 -
Figura V. 70: Diagrama de secuencia de reportes.....	- 129 -
Figura V. 71: Diagrama de Secuencia del objeto evaluación	- 129 -
Figura V. 72: Diagrama de secuencia del objeto reporte evaluación.....	- 130 -
Figura V. 73: Diagrama de secuencia de la obtención de reportes	- 130 -
Figura V. 74: Caso de uso del rol administrador.....	- 131 -
Figura V. 75: Caso de uso del rol maestro	- 131 -
Figura V. 76: Mapa Navegacional	- 133 -
Figura V. 77: Contexto de alumno.....	- 134 -
Figura V. 78: Contexto de curso	- 135 -
Figura V. 79: Contexto de maestro	- 135 -
Figura V. 80: Contexto de paralelo.....	- 136 -
Figura V. 81: Contexto de padres de familia	- 136 -
Figura V. 82: Contexto de materia asignada.....	- 136 -
Figura V. 83: Contexto de login o autenticación	- 137 -
Figura V. 84: Contexto de evaluación.....	- 137 -
Figura V. 85: Contexto de matrícula.....	- 138 -
Figura V. 86: Contexto de registro de pago	- 138 -
Figura V. 87: Contexto de año lectivo	- 138 -
Figura V. 88: Contexto de inscripción	- 139 -
Figura V. 89: Contexto de materia.....	- 139 -
Figura V. 90: Vistas del sistema	- 140 -
Figura II. 91: Pantalla final de inicio de sesión.....	- 140 -
Figura V. 92: Pantalla final de cambio de rol	- 141 -
Figura V. 93: Pantalla final de generación de reporte.....	- 141 -
Figura V. 94: Vistas de Presentación de Reportes	- 143 -
Figura V. 95: Librerías utilizadas para la implemetación.....	- 143 -
Figura II. 96: Desarrollo de clases del sistema	- 144 -
Figura V. 97: Desarrollo de funciones del sistema	- 145 -
Figura V. 98: Desarrollo de controladores del sistema	- 145 -
Figura V. 99: Diseño de Base de Datos	- 146 -
Figura V. 100: Esquema Parvulario.....	- 147 -

ÍNDICE DE TABLAS

Tabla III. I: Parámetros indicadores y métricas de evaluación	- 51 -
Tabla IV. II: Evaluación del tipo de requisitos	- 82 -
Tabla IV. III: Evaluación de la propuesta funcional	- 83 -
Tabla IV. IV: Escala de valoración de la propuesta funcional.....	- 83 -
Tabla IV. V: Comparación según propuesta funcional	- 83 -
Tabla IV. VI: Evaluación del tiempo de diseño	- 85 -
Tabla IV. VII: Escala de valoración para el modelo de producto y proceso.....	- 86 -
Tabla IV. VIII: Evaluación del modelo de producto.....	- 86 -
Tabla IV. IX: Evaluación del modelo de proceso	- 87 -
Tabla IV. X: Evaluación en pesos de puntos de función	- 94 -
Tabla IV. XI: Evaluación de pesos para la comprobación de hipótesis	- 95 -
Tabla IV. XII: Datos para el cálculo de T Student.....	- 97 -
Tabla V. XIII: Descripción caso de uso para autenticación	- 107 -
Tabla V. XIV: Descripción caso de uso para el cambio de rol del administrador	- 108 -
Tabla V. XV: Descripción del caso de uso para el registro de año lectivo	- 108 -
Tabla V. XVI: Descripción de caso de uso de inscripción.....	- 109 -
Tabla V. XVII: Descripción de caso de uso del registro de alumno	- 109 -
Tabla V. XVIII: Descripción de caso de uso de matrícula.....	- 110 -
Tabla V. XIX: Descripción de caso de uso del registro de maestro.....	- 110 -
Tabla V. XX: Descripción de caso de uso de padres de familia	- 111 -
Tabla V. XXI: Descripción de caso de uso del representante	- 111 -
Tabla V. XXII: Descripción de caso de uso de cursos y paralelos	- 111 -
Tabla V. XXIII: Descripción de caso de uso de materia.....	- 112 -
Tabla V. XXIV: Descripción de caso de uso de registro de pago.....	- 112 -
Tabla V. XXV: Descripción de caso de uso de evaluaciones a alumnos	- 113 -
Tabla V. XXVI: Descripción de caso de uso de crear nuevas cuentas	- 113 -
Tabla V. XXVII: Descripción del caso de uso de reporte de lista de alumno.....	- 114 -
Tabla V. XXVIII: Descripción de caso de uso del reporte de registro de alumno.....	- 114 -
Tabla V. XXIX: Descripción de caso de uso del reporte de registro de pagos por alumno..	- 115 -
Tabla V. XXX: Descripción de caso de uso del reporte orden de pago.....	- 115 -
Tabla V. XXXI: Definición de roles	- 132 -

Tabla V. XXXII: Descripción de datos de la tabla anio_lectivo.....	- 147 -
Tabla V. XXXIII: Restricción de la tabla anio_lectivo.....	- 148 -
Tabla V. XXXIV: Descripción de datos de la tabla alumno.....	- 148 -
Tabla V. XXXV: Restricción de la tabla alumno.....	- 149 -
Tabla V. XXXVI: Descripción de datos de la tabla inscripción.....	- 149 -
Tabla V. XXXVII: Restricciones de la tabla inscripción.....	- 149 -
Tabla V. XXXVIII: Descripción de datos de la tabla matrícula.....	- 150 -
Tabla V. XXXIX: Restricciones de tabla matricula.....	- 150 -
Tabla V. XL: Descripción de datos de tabla persona.....	- 151 -
Tabla V. XLI: Restricción de tabla persona.....	- 151 -
Tabla V. XLII: Descripción de datos de la tabla padre.....	- 152 -
Tabla V. XLIII: Restricciones de tabla padres_familia.....	- 152 -
Tabla V. XLIV: Descripción de datos de la tabla padre_alumno.....	- 152 -
Tabla V. XLV: Restricciones de tabla padre_alumno.....	- 153 -
Tabla V. XLVI: Descripción de datos de la tabla representante.....	- 153 -
Tabla V. XLVII: Restricciones de tabla representante.....	- 153 -
Tabla V. XLVIII: Descripción de datos de la tabla profesor.....	- 154 -
Tabla V. XLIX: Restricciones de tabla profesor.....	- 154 -
Tabla V. L: Descripción de datos de la tabla representantepoli.....	- 155 -
Tabla V. LI: Restricciones de la tabla representantepoli.....	- 155 -
Tabla V. LII: Descripción de datos de la tabla curso.....	- 156 -
Tabla V. LIII: Restricción de la tabla curso.....	- 156 -
Tabla V. LIV: Descripción de datos de la tabla paralelo.....	- 156 -
Tabla V. LV: Restricciones de la tabla paralelo.....	- 157 -
Tabla V. LVI: Descripción de datos de la tabla materia.....	- 157 -
Tabla V. LVII: Restricción de tabla materia.....	- 157 -
Tabla V. LVIII: Descripción de datos de la tabla materia asignada.....	- 158 -
Tabla V. LIX: Restricciones de la tabla materia_asignada.....	- 158 -
Tabla V. LX: Descripción de datos de la tabla evaluación.....	- 159 -
Tabla V. LXI: Restricciones de la tabla evaluación.....	- 159 -
Tabla V. LXII: Descripción de datos de la tabla registro pago.....	- 159 -
Tabla V. LXIII: Restricción de la tabla registro_pago.....	- 160 -
Tabla V. LXIV: Descripción de datos de la tabla costo.....	- 160 -
Tabla V. LXV: Restricción de la tabla costo.....	- 161 -

INTRODUCCION

En el ámbito de las metodologías de desarrollo de software, se aprecia una mejora en la puesta en práctica de dichas metodologías de desarrollo, así como la flexibilización de éstas para potenciar la productividad de las mismas sin renunciar a la calidad.

La mejora de la efectividad y la productividad en el desarrollo de software está ligada a la utilización de buenas prácticas de Ingeniería de Software. En la actualidad es indiscutible que el uso de una metodología apropiada es un factor clave para el éxito de cualquier esfuerzo de ingeniería. La ingeniería de software, por su relativa juventud como disciplina y por la altísima variabilidad de los productos que gestiona, pocas organizaciones que desarrollen software utilizan metodologías de forma sistemática, sin embargo la Ingeniería de procesos contribuye en esta línea, diseñando y construyendo metodologías en función de las necesidades específicas de cada organización.

De este modo, que las metodologías deben responder a multiplicidad de estándares, también deben adaptarse a las características particulares de cada uno de los proyectos que se llevan a cabo en la organización. La complejidad del proceso hace imprescindible una evaluación de ellas en base a ciertos parámetros e indicadores, el cual mediante la ayuda del desarrollo de prototipos se realiza la elección de la metodología adecuada.

CAPÍTULO I

MARCO REFERENCIAL

En el presente capítulo se describe el contenido del marco de referencia que se encuentra constituido por: problematización, justificación, objetivos del proyecto y recursos.

1.1. ANTECEDENTES

La web se ha convertido en un instrumento de uso cotidiano para el intercambio de información en nuestra sociedad. En 1990, Tim Berners-Lee presento un proyecto de “World Wide Web” en el CERN (Suiza), constituyéndose como uno de los medios de publicación más importantes. La arquitectura inicial de la Web está basada en tres pilares fundamentales: HTML, URLs y HTTP; de acuerdo a su evolución estos componentes han cambiado de manera que involucre la interacción, seguridad y servicios web.

Recordando un tiempo atrás dentro del mercado de contenidos en la web, se tiene que a partir de 1994 empezaron a originarse las páginas iniciales. Con la presentación de la conferencia de Texas “Hipertext – 1991” se dio la primera demostración, que fue realizada en Programación HTML con archivos de texto y/o información sobre bases de datos enlazadas con el hipertexto, cuadros y vínculos; la mayoría desarrolladas por programadores con poca habilidad en diseño

gráfico y comunicación, resultaban páginas simples sin definiciones de comunicación ni decisiones estéticas de jerarquía; lo que queda en la historia como los primeros en imponer un nuevo camino, sin embargo la investigación realizada para presentar la conferencia tuvo que ser expuesta por varios análisis fallidos. (Del Valle Rodríguez, 2009)

El crecimiento continuo de las tecnologías y la implementación de las mismas da origen a la Ingeniería web que busca procedimientos que permitan un proceso incremental dentro de la ingeniería de software proporcionando herramientas de análisis y diseño.

En 1998, Roger Pressman moderó una mesa redonda virtual con representantes de la ingeniería de software tradicional y del desarrollo de software basado exclusivamente en la web. La idea general fue que aplicar un proceso de ingeniería nunca es una mala idea pero que éste debería adaptarse a los requerimientos de cambio continuo y rapidez, siempre presentes, en el proceso de desarrollo Web. (Pressman R. S. & Troya J. M., 1988)

El desarrollo de Aplicaciones Web (WebApp) es distinto respecto al desarrollo de software de otras categorías, estas diferencias son planteadas por Powell y referenciadas en Pressman en el año 2003: “lo que implica la realización de una mezcla de publicación impresa y desarrollo de software (programación), marketing e informática, de comunicaciones internas y relaciones externas, también la influencia del arte y el uso de nueva tecnología”; añadiendo el desarrollo de la navegación y presentación de la aplicación web.

La ingeniería web propone las primeras bases en los años 90 que incluyen conceptos de navegación, pero para la segunda generación se definen los primeros términos de metodología, enfocándose hacia el uso de estándares, procesos y herramientas, (Concepción, 2010). En la actualidad existen varios tipos de aplicaciones web que están orientados a la información, interacción con el usuario, servicios, entornos colaborativos y más, lo que nos lleva a definir diferentes metodologías usadas en ingeniería web dentro de los cuales tenemos: Modelo de Diseño de Hipertexto (HDM), Metodología de Administración Relacional (RMM), Método de

Diseño de Hipermedia Orientado a Objetos (OOHDM), Solución Web Orientado a Objetos (OOWS), UML basado en Ingeniería Web (UWE), EORM (Metodología de Relación entre Objeto, RNA: Relationship-Navegational Analysis, HFPM: Hypermedia Flexible ProcessModeling. (Del Valle Rodríguez, 2009)

“La no aplicación de una metodología apropiada para el desarrollo web, que cumpla con la navegabilidad, usabilidad e interacción del usuario, ya que los desarrolladores actuales utilizan metodologías de software generales que no son orientadas a la web.”

Para la sistematización del problema se propone establecer una investigación que confronte el problema de no contar con una metodología apropiada para el diseño de una aplicación web, el cual es necesario realizar las siguientes preguntas:

¿Qué es Ingeniería Web?, ¿Qué es aplicación web, y que tipos existe?, ¿Qué metodologías de diseño existe en Ingeniería Web?, ¿Cuántas metodologías existen para el diseño?, ¿Qué ventajas proporciona a una aplicación que utiliza metodología web, ¿Qué metodología usa menos recursos y brinda grandes ventajas?.

1.2. JUSTIFICACIÓN DEL TEMA

Se describe la razón o el porque de la investigación, tanto teórico, metodológico y práctico.

1.2.1. JUSTIFICACIÓN TEÓRICA

Con el avance tecnológico implementado en las diferentes empresas y centros educativos en la actualidad, se ha detectado la necesidad de realizar una aplicación web, con la ayuda de un software específico partiendo del paradigma de Ingeniería web que involucra elementos de usabilidad, navegabilidad, funcionalidad en las aplicaciones, a diferencia de la ingeniería de software.

Se conoce la existencia de varias metodologías web, sin embargo al desarrollar una aplicación no siempre se cuenta con una metodología adecuada a pesar de su existencia, es por este motivo que el presente proyecto de tesis pretende analizar y comparar entre las metodologías UWE y OOWS, la más apropiada para obtener una mejor productividad en el desarrollo de la aplicación web dirigida al Centro Parvulario Politécnico.

1.2.2. JUSTIFICACIÓN METODOLÓGICA

Para el desarrollo del proyecto propuesto se establece dos tipos: de investigación y caso práctico; razón por la cual para el proceso de investigación se utilizará el método científico y para el desarrollo de la aplicación web, la metodología web seleccionada de la comparación realizada, las metodologías UWE: UML basado en Ingeniería Web, y OOWS: Solución Web Orientado a Objetos, establecidas para la comparación y para el desarrollo del caso práctico, en el cual todas y cada una de las etapas o fases que de la metodología seleccionada será aplicada.

1.2.3. JUSTIFICACIÓN PRÁCTICA

Mediante la implementación de una aplicación web en el centro Parvulario Politécnico se pretende disminuir el trabajo administrativo, incorporando las nuevas tecnologías con el objetivo de facilitar las tareas del administrador con la ayuda de una aplicación web. Cabe recalcar que las tareas fundamentales a desarrollarse involucra el ingreso y actualización de datos de alumnos, maestros, materias, usuarios y el registro de inscripción, matrícula, evaluaciones y pago de aranceles, especificando también que el acceso a dicha aplicación será exclusivo para el personal administrativo por medio de roles de usuario.

Al conocer el proceso de desarrollo de las metodologías propuestas, se debe seleccionar la más apropiada en cuanto a productividad de la aplicación web, es por este motivo que se realizara prototipos con cada una de dichas metodologías para realizar un estudio comparativo entre ellas.

Este trabajo de investigación, en la ESPOCH se enmarca en la LINEA: V. Tecnologías de la información, comunicación y procesos industriales; con el Programa: para el desarrollo de aplicaciones de software para procesos de gestión y administración pública y privada; y dentro del Plan Nacional del Buen Vivir con la Política 2.7. Promover el acceso a la información y a las nuevas tecnologías de la información y comunicación para incorporar a la población a la sociedad de la información y fortalecer el ejercicio de la ciudadanía.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Realizar un estudio comparativo entre las metodologías UWE y OOWS para mejorar la productividad en el desarrollo de aplicaciones web.

1.3.2. OBJETIVOS ESPECÍFICOS

1. Estudiar las metodologías UWE y OOWS utilizadas en Ingeniería Web para el desarrollo de aplicaciones web.
2. Desarrollar prototipos de un módulo específico aplicando las metodologías estudiadas, utilizando el lenguaje de programación de Java JSF.
3. Establecer los parámetros de medición que determinen la mejor productividad en el desarrollo de aplicaciones.
4. Realizar un estudio comparativo de las metodologías UWE y OOWS para seleccionar la metodología que mejore la productividad en el desarrollo de aplicaciones web.
5. Desarrollar una aplicación web en el centro Parvulario Politécnico, aplicando la metodología seleccionada en el estudio.

1.4. RECURSOS

Se define las herramientas software y equipos hardware utilizados para la investigación.

Herramientas Software

- DBMS PostgreSQL 9.2: Motor de Base de datos bajo licencia BSD usado para la creación de las tablas, funciones y dominios necesarios para la aplicación.
- Enterprise Architect 8: Herramienta CASE de diseño utilizada para el desarrollo de diagramas.
- Plataforma NetBeans: Es el IDE de desarrollo base usado para la creación de la aplicación.
- Java Prime Face IDE 7.3: Es el componente usado para la creación de las vistas de la aplicación.
- Cocomo: Herramienta de estimación de costos.
- Servidor Glassfish 3.1.1: Permite ejecutar aplicaciones que implementa las tecnologías definidas en la plataforma Java EE.
- JasperReports 4.7.1: Es la herramienta utilizada para la creación y presentación de los reportes

Equipos Hardware

- Computador HP G42-270LA (procesador Intel CORE i3, disco de 500 GB y memoria de 4 GB)

CAPÍTULO II

MARCO TEÓRICO

A continuación se presenta el sustento teórico acerca de los conceptos de la Ingeniería Web y las metodologías que ayudan al diseño de aplicaciones.

2.1. INGENIERÍA WEB

Según Adam Smith define como un “paradigma nuevo que contiene conjunto compartido de suposiciones que busca el logro de habilidades para asumir el futuro y el cambio que Internet impone, y nos ayuda a predecir su comportamiento”.

Según (Del Valle Rodríguez, 2009) “La Ingeniería Web (IWeb) que aplica sólidos principios científicos, de ingeniería y de administración, con enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en Web de alta calidad”, en el cual se puede manifestar que ambos autores coinciden en el enfoque de lograr un producto de calidad y alto rendimiento hacia el futuro por medio del internet. Pero difieren en la conceptualización de los principios, ya que para uno se enfoca en las tecnologías web, mientras que para el otro involucra la ingeniería y la administración.

Uno de los objetivos principales de la Ingeniería Web es la aplicación y desarrollo de los diferentes enfoques que aplica a la Ingeniería de Software común para llegar al desarrollo pleno y de calidad de las aplicaciones Web. Existen muchas características que ubican a dichas aplicaciones en una posición especial dentro de la Ingeniería de Software. El cual se puede mencionar las siguientes;

- El contenido de calidad y la presentación Web que permiten la navegabilidad correcta hacen necesaria una precisa colaboración dentro de un equipo de trabajo.
- Es importante mencionar que obtener los requerimientos funcionales implica complejidad, ya que hay que considerar la calidad del contenido a la hora de desarrollar acciones que permitan satisfacer al usuario, (Loucopoulos & Karakostas, 1995).
- La determinación de perfil de usuarios conlleva un contexto social que se torna extremadamente complejo, pero no imposible determinarlo; ya que existen formas y estrategias educativas y culturales que hacen que los parámetros de usabilidad se vuelvan relativos a la hora del diseño, (Oliveros, A., et al Wehbe, R., Del Valle Rojo, S., & Rousselot, J., 2011).
- La diversidad de plataformas existentes para la accesibilidad y navegación de aplicaciones web va cambiando el panorama común por la especial interacción que presenta entre los requerimientos y la arquitectura.

Para establecer los fundamentos de la ingeniería web es necesario mencionar los principios que se estudia, como son ontologías que incluyen definiciones tecnológicas, la web semántica que interpreta al internet como una manera rápida de encontrar respuestas y agentes de software que funciona como un proceso computacional que provee el aprendizaje el cual incluye dos procesos: uno de interacción externo, que ocurre entre el aprendiz y su entorno social y cultural; y otro psicológico interno en el que nuevos impulsos son conectados con resultados de aprendizaje anteriores.

En el proceso de desarrollo de un sistema, sea o no para la web, el equipo de desarrollo se enfrenta al problema de la identificación de requisitos. La definición de las necesidades del sistema es un proceso complejo, pues en él hay que identificar los múltiples requisitos que el sistema debe cumplir al momento del desarrollo el cual debe satisfacer las necesidades de clientes y usuarios finales.

Para realizar este proceso, no existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado. Existe en cambio un conjunto de técnicas, cuyo uso proponen las diferentes metodologías para el desarrollo de aplicaciones web. Se debe tener en cuenta que la selección de las técnicas y el éxito de los resultados que se obtengan, depende en gran medida tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participen.

Este escenario motiva, en 1998, a San Murugesan, Athula Ginige, Yogesh Deshpande y Steve Hansen a establecer una nueva disciplina: la Ingeniería Web, definiéndola de la siguiente manera:

“La Ingeniería Web maneja el establecimiento y uso de principios sensatos científicos, de ingeniería y gestión, así como con enfoques disciplinados y sistemáticos, para el desarrollo, instalación y mantenimiento correcto, de aplicaciones y sistemas de alta calidad, basados en la Web”, (Quintero, 2008)

2.2. INGENIERÍA DE REQUISITOS EN LA INGENIERÍA WEB

Dentro de la Ingeniería de Requisitos el principal problema es la presentación de los requerimientos en lenguaje común. Investigaciones recientes demuestran que las herramientas y documentos utilizados para el análisis de requerimientos son suministrados por los mismos usuarios; es decir exponen las necesidades que requieren que realice la aplicación mediante la expresión de líneas textuales o gráficos.

La definición de los requerimientos que la aplicación debe cumplir es un proceso amplio y complejo, pues en él hay que identificar las necesidades que el usuario final debe cumplir mediante el uso del sistema, el cual facilite el trabajo diario de los diferentes clientes.

Para realizar este proceso, no existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado. Existe en cambio un conjunto de técnicas, cuyo uso proponen las diferentes metodologías para el desarrollo de aplicaciones web. (Mich, et al Franch, Mariangela, & Novi, I., 2004) Se debe tener en cuenta que la selección de las técnicas y el éxito de los resultados que se obtengan, depende en gran medida tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participen.

Para facilitar la comprensión de las propuestas, antes de presentarlas, presentamos una clasificación de requisitos relevantes en sistemas web (Escalona & Koch, 2002).

- **Requisitos de datos**, llamados requisitos de contenido porque la información que abarca es básica y fundamental, requisitos conceptuales o de almacenamiento de información, denominados así porque responden a la pregunta: ¿qué datos debe guardar y administrar el sistema?.
- **Requisitos de interfaz**, estos requisitos van dirigidos al usuario, llamados así por la interacción que presenta con el usuario. Es la forma de expresar cómo va a interactuar el usuario con el computador mediante interfaces que presenta el sistema.
- **Requisitos navegacionales**, presentan la facilidad de desplazarse o navegar dentro del sistema mediante la determinación de las necesidades de navegación del usuario.
- **Requisitos de personalización**, ayuda a la descripción de como el usuario debe adaptarse al sistema mediante la interacción de los mismos, ya que cada usuario tiene su personalidad y preferencias.
- **Requisitos transaccionales o funcionales**, buscan lo el sistema debe realizar internamente, acciones de programación o conocidos también como transacciones de

código que ayudan a cumplir con los requisitos de servicio, sin tomar en cuenta aspectos de interfaz o interacción.

- **Requisitos no funcionales**, denominados así por ser requisitos complementarios para el funcionamiento como el entorno de desarrollo, portabilidad, disponibilidad, flexibilidad y otros no mencionados pero que ayudan al desempeño del sistema.

2.3. APLICACIÓN WEB

Una aplicación Web es un sistema de software al que se accede a través de Internet, las aplicaciones Web constituyen una clase especial de aplicaciones de software que se construyen de acuerdo con ciertas tecnologías y estándares relacionando con la accesibilidad, usabilidad, navegabilidad y presentación, de manera que el usuario tenga una mayor efectividad en su aplicación. (Kappel, et al Pröll, B., Reich, S., & Retschitzegger, W., 2006)

2.3.1. CATEGORÍA DE APLICACIONES WEB

Varios autores (Pressman R. S. & Troya J. M., 1988) han definido taxonomías para tratar de clasificar los distintos tipos de aplicaciones web. La más conocida y utilizada en el ámbito de la ingeniería es la de Ginige y Murugesan definiendo las siguientes categorías de aplicaciones web (Moreno L., 2011) : de información (noticias en línea, servicios de noticias, catálogos, manuales), interactivas (formularios de registro, servicios en línea), transaccionales (compras electrónicas, bancos en línea), sistemas de *workflow* (planificación en línea, administración de inventarios, monitorización de estados), ambientes de trabajo colaborativo (sistemas distribuidos de autoría, herramientas colaborativas), comunidades en línea (grupos de *chat*, sistemas de recomendación) y portales web (centros comerciales electrónicos, intermediarios en línea). Sin embargo de acuerdo a (Kappel et al., 2006) las aplicaciones web se clasifican por su complejidad como de muestra en la figura II.1.

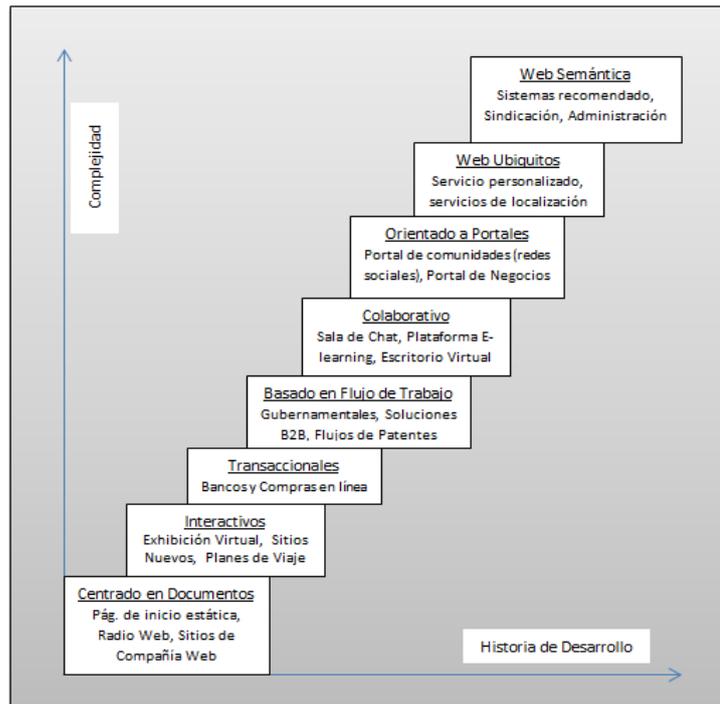


Figura II. 1: Clasificación de Aplicaciones

Fuente: Tesis, AWA (Moreno L., 2011)

2.4. DEFINICIÓN DE METODOLOGÍA WEB

El uso de metodologías web parte de las necesidades de evolución en procesos creativos, mantenimiento, la adaptación a nuevos dispositivos de acceso y la migración a nuevas plataformas y entornos de desarrollo que deben encaminarse con la ayuda de un estándar o modelo de desarrollo.

El desarrollo de un WebApp es un esfuerzo multidisciplinario debido al manejo de múltiples formatos, con una mezcla de atención social para saber su efecto ético y legal. Los primeros años que se incursionó en el desarrollo de la Web, las aplicaciones eran caóticas y carecían de estructura, lo que evidenciaba la falta de métodos de desarrollo adecuados, por ello se ha venido impulsado en el uso de metodologías web, definidas como un proceso estructurado y sistemático que ayuda a generar una solución informática dentro de la web.

Para todo esto se han desarrollado metodologías que permiten estructurar comunicar, entender, simplificar y formalizar tanto el dominio como las decisiones de diseño, así como disponer de documentación detallada para posibles cambios del software.

2.4.1. EVOLUCIÓN DE LAS METODOLOGÍAS DE DESARROLLO WEB.

En la historia el desarrollo de las distintas metodologías ha llevado a determinar una evolución de las mismas presentadas en generaciones en base a su sofisticación, (Duque Escobar, 2011) se menciona tres y son las siguientes:

Primera Generación: Inicio a principios de los 90, se crean las bases de la ingeniería Web, en los que se incluyen conceptos como desarrollo de la navegación, separación entre estructuras y el contenido durante el ciclo de desarrollo, mediante fases de diseño y fases de programación.

Segunda Generación: Para continuar la siguiente generación se toma desde la segunda mitad de los 90, en el cual se refinan los primeros modelos y se añaden los soportes de funcionalidad básica; también se llevan a cabo la presentación de los primeros bosquejos o esquemas del proceso donde se delimitan los modelos conceptual, lógico y físico.

Tercera generación: (A partir del 2000): Se lleva a cabo la profundización en el soporte para la funcionalidad, enfatización de la figura del usuario en los métodos, y se avanza hacia la estandarización de notaciones, procesos y lenguajes de especificación.

2.4.2. TIPOS DE METODOLOGÍAS WEB

Los Métodos de Ingeniería Web detectaron la necesidad de tener técnicas de modelado para los ambientes web que modelen la presentación, contenido y navegación. Por ello crearon técnicas compuestas de constructores con los que crear elementos de modelado, reglas para construir modelos, y transformaciones automatizables hasta crear aplicaciones.

En la figura II.2, se muestra el panorama de los métodos existentes en el ámbito de Ingeniería Web, indicando cuándo surgieron y en qué técnicas no específicas para la Web se basan (ER,

OMT, UML), (Escalona & Koch, 2002) y cómo han ido evolucionando. Cada método tiene un enfoque: algunos basados en cómo administrar sus datos, otros con el foco en el hipertexto.

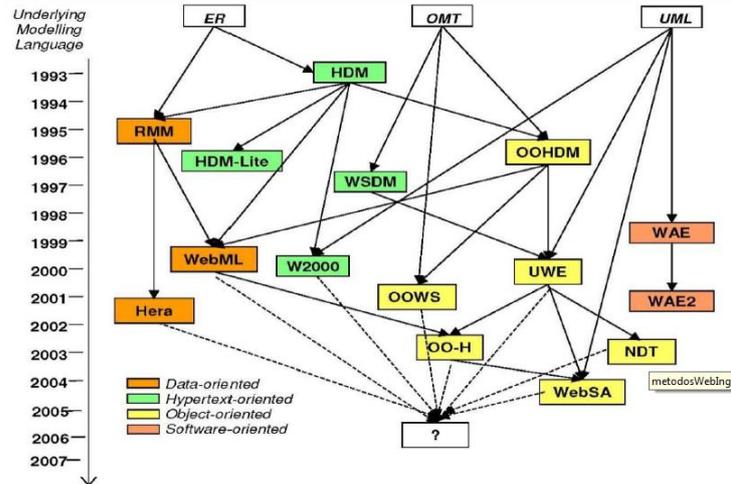


Figura II. 2: Métodos de Ingeniería Web

Fuente: Artículo Científico , Web Engineering (Kappel et al., 2006)

En los últimos años aparecen un conjunto de métodos para desarrollar aplicaciones web en donde no toman en cuenta los principios básicos de programación, ni tampoco los que son proporcionados por la ingeniería web. Estos métodos presentan la integración de las actividades técnicas y gerenciales exponiendo así en forma explícita su modelo de proceso, lo que garantiza el desarrollo de una aplicación web de calidad, y de manera implícita describen el producto, es decir las características arquitecturales de las aplicaciones web, el análisis previo y diseño de diagramas respectivos.

Los métodos que a continuación se presentan se han agrupado de acuerdo a su modelo de proceso y al contexto particular donde pueda ser aplicado. Por esta razón se clasifican de la siguiente manera:

Procesos Ágiles para el desarrollo de aplicaciones

Estos métodos están caracterizados por poseer iteraciones cortas, pruebas continuas y constante colaboración, y son los siguientes: (Extreme Programming) XP: Programación extrema, Open

Source: Código abierto (Caceres y Marcos, 2002), Dynamic Systems Development Method (DSDM): Metodo de desarrollo de sistemas dinámicos.

Métodos para el desarrollo de sistemas de información web (SIW)

Se definen por seguir una secuencia ordenada de pasos requeridos para el desarrollo y emplean técnicas de análisis y diseño orientado a objetos, entre los más conocidos están: Metodología para desarrollar Sistemas de Información (MIDAS)(Cáceres y Marcos, 2001) y el método para desarrollar SIW (Torralba , J., Cuenca, J., & Ruíz , V, 2003).

Métodos para el desarrollo de aplicaciones hipermedia.

Estos métodos solo dan especial importancia al diseño y no complementa el ciclo normal de desarrollo, como son: The object Oriented Hypermedia Design Model (OOHDM): Modelo de diseño hipermedial orientado a objetos, Relationship Management Methodogy (RMM): Metodología de administración relacional, Hypermedia Flexible ProcessModeling (HFPM): Modelo de Proceso Flexible Hypermedial (Costagliola, Feruucci y Francese, 2002).

Métodos para el desarrollo se sitios web instruccionales.

Este tipo de métodos posee varios componentes organizacionales, administrativos, instruccionales y tecnológicos con el objetivo de garantizar la calidad del producto, entre ellos están: Desarrollo de sitios web instruccionales (Montilva y Barrios, 2002), Simple Web Method (SWM): Método web simple (Cormack, Griffiths, 2002), y el Modelado web basado en un sistema instruccional.

Métodos para el desarrollo de aplicaciones de comercio electrónico: (e-commerce).

El más usado de este tipo esta el Marco de Referencia Basado en componentes para e-commerce.

Metodologías para el desarrollo de aplicaciones web.

La característica fundamental para estos métodos es que tiene ciclos de desarrollo cortos, enfocándose en la forma del contenido y el diseño de la estructura de navegación, de este tipo se tiene los siguientes: Web Site Design Method (WSDM): Método de diseño de Sitio web (Escalona & Koch, 2002), Relationship-NavigationalAnálisis (RNA): Análisis Navegacional-Relacional, UML-based Web Engineering Approach (UWE): UML basado en Ingeniería web (Koch, 2001), y OOWS: Método de Producción de software en ambientes Web: Object-Oriented Web Solution , (Pastor, A. & Fons Joan, 2002).

2.4.3. VENTAJAS DE UTILIZAR UNA METODOLOGIA

Se ha llevado a cabo una investigación en los distintos métodos, para analizar las ventajas y desventajas en relación de cómo poder integrar la accesibilidad web de manera sistemática con el soporte que puedan ofrecer. Se ha detectado, en general, un apoyo pobre en requisitos y casi todos los métodos están muy centrados en la arquitectura, (Escalona & Koch, 2002). También se han detectado problemas en herramientas para construir modelos y generar las aplicaciones, pues son específicas de cada uno de los métodos.

En el ámbito de la Ingeniería Web, la accesibilidad es referida, aunque a veces de manera muy superficial, tratándose como parte de la usabilidad y también hay trabajos que la consideran como una característica independiente en la Ingeniería Web. Son varios los proyectos que incluyen accesibilidad: algunos con el uso de patrones sobre un Método de Ingeniería Web para incluir algunos requisitos de accesibilidad en interfaces de usuario, o el Método WebML, que trata sólo la dimensión de la accesibilidad cercana a la usabilidad, centrándose en el uso fácil y eficaz en la recuperación y la navegación de la información para una aplicación web, pero no cómo llevar a cabo la accesibilidad de acuerdo con estándares como WCAG teniendo en cuenta, únicamente algunos aspectos de la accesibilidad a ofrecer versiones de texto como alternativa para los contenidos con imágenes (Moreno L., 2011).

2.5. UWE: UML BASED WEB ENGINEERING

La Ingeniería Web basada en UML, es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web (Rossainz & Ocampo, 2001).

UWE se define también como una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está diseñado como una extensión conservadora del modelo UML (versión 1.4), conservando muchos de sus elementos que no ha sido modificados y añadiendo características o asociaciones para el modelado de elementos. Para ello se define las relaciones y características adicionales para otros elementos modeladores y restricciones de uso OCL para especificar lo adicional que es la semántica estática.

2.5.1. PROCESO O FASES

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de usuario, modelo de adaptación y modelo de tareas para representar los aspectos dinámicos de la aplicación mediante la descripción de situaciones. De esta manera se obtiene una colección de modelos y diagramas que describen una aplicación Web de manera integral. (Kraus & Koch, 2002)

Análisis de requisitos: El análisis de requisitos se expresa a través de la especificación de los casos de uso del sistema. Un caso de uso en UML es una unidad coherente de la funcionalidad proporcionada por la aplicación que obra recíprocamente con unos o más actores de la aplicación. Describe una parte del comportamiento de la aplicación sin revelar la estructura

interna. De esta manera, los requisitos para una aplicación Web se pueden especificar con un modelo de casos de uso.

UWE clasifica los requisitos en dos grandes grupos: funcionales y no funcionales. Para la determinación de requisitos funcionales para esta metodología se relaciona los siguientes:

- Requisitos de contenido, registro de información.
- Requisitos de estructura, como está conformado.
- Requisitos relacionados con la presentación, de interfaz.
- Requisitos relacionados con la adaptación, refinación del modelo conceptual.
- Requisitos relacionados con los usuarios, identificación de roles.

Además, UWE propone como técnicas apropiadas para la captura de los requisitos de un sistema web las entrevistas, los cuestionarios, los checklists y los casos de uso, los escenarios y el glosario para la definición de requisitos.

La estructura de los paquetes dentro de UWE depende de los diagramas UML correspondiente al paquete del más alto nivel como se observa en la figura II.3.

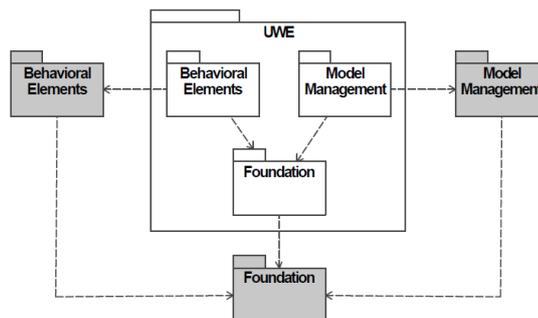


Figura II. 3: Paquetes del Nivel alto

Fuente: Artículo “A metamodel for UWE” (Kraus & Koch, 2002)

Diseño conceptual: El modelado conceptual para aplicaciones de Web dentro de UWE no difiere del modelado conceptual para aplicaciones normales. Sin embargo se introduce algo

nuevo la clase `ConceptualClass` que es heredada de la Clase del elemento UML pero no tiene características adicionales.

En esta etapa se representa el dominio del problema con un diagrama de clases de UML. Los casos de uso sirven como entrada para elaborar tarjetas Clase-Responsabilidad-Colaborador (CRC), o para la identificación de verbos y sustantivos, entre otras técnicas, que permiten determinar las clases, métodos y atributos como explica en la figura II.4.

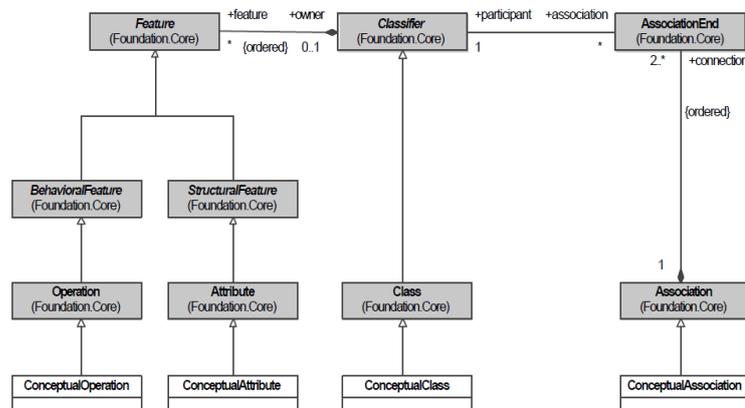


Figura II. 4: Modelado Conceptual

Fuente: Artículo “A metamodel for UWE” (Kraus & Koch, 2002)

Diseño navegacional: El modelo de procesos en UWE posee tres vistas:

- (1) una vista de integración con el modelo navegacional
- (2) una vista estructural y
- (3) una vista de comportamiento o modelo de flujo del proceso.

El propósito de las vistas estructural y de flujo del proceso – según (Kraus & Koch, 2002) UWE es “*modelar los procesos en sí mismos de forma independiente al modelado de la navegación, buscando con ello la separación de aspectos*”.

Los elementos básicos en el modelo de navegación son nodos y enlaces. El modelado correspondiente a los elementos UWE son `NavigationNode` y `Link` que se derivó de la Claves UML y Agrupación, respectivamente.

El diseño de las estructuras de navegación establece las estructuras de acceso que permiten visitar los objetos del espacio navegacional. Están constituidas por menús, índices, visitas guiadas, y formularios.

Diseño de la presentación: El modelo de presentación en UWE está muy relacionado con los elementos de las interfaces definidas en HTML. Estos elementos también están definidos como estereotipos de UML. Los elementos del modelo de presentación son: ventanas, entradas de texto, imágenes, audio y botones. La estructura de presentación es modelada con clases y subclases especializadas que contiene toda la lógica interfaz de usuario y la forma de navegación mediante las clases respectivas indicado en la figura II.5.

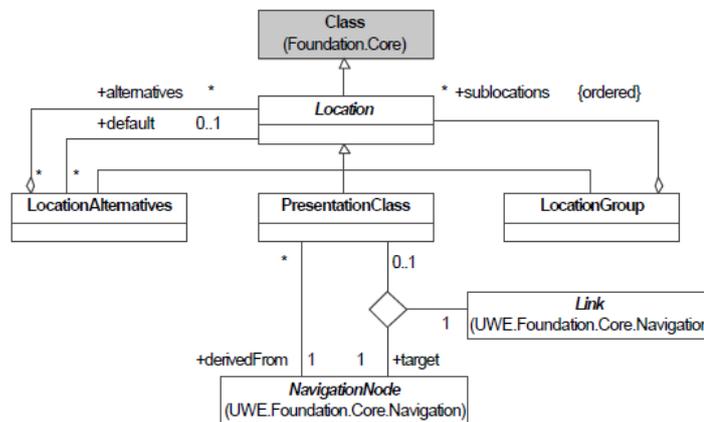


Figura II. 5: Modelo de Presentación UWE

Fuente: Artículo “A metamodel for UWE” (Kraus & Koch, 2002)

Escenarios Web.- Un diagrama de estados de UML denota una secuencia de los estados que un objeto puede adquirir durante su vida, junto con acciones responsivas, disparando eventos y las condiciones asociadas para indicar transiciones. UWE da otro sentido a los diagramas de estados del UML puro ya que los utiliza para visualizar escenarios de navegación. Estos diagramas permiten detallar la parte dinámica del modelo de navegación, especificando los eventos que disparan las situaciones, definen condiciones y explícitamente incluyen las acciones que son realizadas. Junto con el modelo de interacción temporal, los escenarios Web proveen la representación funcional dinámica del modelo de navegación.(Kraus & Koch, 2002)

Modelos Complementarios

Estos modelos agregan varios diagramas que ayudan a la refinación del diseño en la arquitectura.

Modelo de Usuario: El elemento básico de este modelo es el usuario que se implementa en el diagrama de Casos de Uso. Cada usuario tiene una identificación única, y es un elemento importante en el modelo de adaptación llamado UserProfile que puede ser asignado ya sea para un User Role para un grupo de usuarios o para un usuario personal. Administra propiedades independientes aplicativas como el nombre de usuario y otros atributos; por otra parte se distingue entre los diferentes modelos de aplicaciones de Web como lo conceptual, la navegación y el aspecto de presentación.

Modelo de Adaptación: Los elementos básicos en los modelos de adaptación son las reglas y los acontecimientos que provocan estas reglas.

Las reglas de adaptación son clases relacionadas con propiedades del usuario y a las claves de lo conceptual, la navegación y los paquetes de presentación.

Modelo de Tareas: hace dos propuestas diferentes: Una extensión UML de casos de uso y otro basaron en los diagramas de actividad. Los casos de uso del sistema se consideran como tareas en nivel de análisis y los de diagramas de actividad como "mapas de carreteras" de sistema comportamiento funcional.

2.5.2. VENTAJAS Y DESVENTAJAS

Ventajas

- Es una extensión basada en los mecanismos definido por UML con la ventaja de usar una notación estándar que puede ser fácilmente mantenido por el usuario y eso no le impacta los formatos de intercambio.

- El perfil UWE incluye estereotipos y valores etiquetados definidos para los elementos modeladores de aplicaciones de Web, como la navegación, la presentación, el usuario, la tarea y aspectos de adaptación.
- Todos los nuevos elementos de modelado de UWE se relacionan por herencia para al menos un elemento de UML.
- La ventaja es crear fácilmente lenguajes de modelado para aplicaciones de dominios específicos con elementos de modelado más expresivos y claramente definidos.

Desventajas

- El riesgo es que el excesivo uso de estereotipos puede hacer a un lenguaje, difícil de manejar y entender aspectos de adaptación.
- La inclusión en el modelo navegacional de la clase de procesos como una primitiva que posee una semántica distinta a los requisitos que se esperan capturar en este modelo podría dificultar la comprensión del mismo.
- No existe integración de funcionalidad disponible en otras aplicaciones, los procesos se construyen a partir de funcionalidad propia y no se consume funcionalidad ajena disponible en otras aplicaciones.
- No existe tampoco producción de funcionalidad que pueda consumirse por otras aplicaciones, dificultando con ello la definición de procesos Negocio-a-Negocio.

2.6. OOWS: OBJECT ORIENTED WEB SOLUTION

El método OOWS es la extensión para el modelado conceptual de aplicaciones Web del método de producción de software OO-Method (Pastor, A. & Fons Joan, 2002). Los requisitos para este tipo de aplicaciones se capturan en un par de modelos -de *Navegación* y *Presentación*- que se agregan a los ya existentes en OO-Method: el *Modelo Estructural* (Diagrama de clases), el *Modelo Dinámico* (Diagramas de Estado y Secuencia) y el *Modelo Funcional* (Especificación textual tomado a partir de la especificación formal OASIS). (Ocampo & Ramón, 2009)

2.6.1. PROCESO

Esta metodología posee una arquitectura específica de funcionamiento en el cual contiene el módulo de Especificación del Sistema y el módulo de Desarrollo de la Solución como se indica en la figura II.6.

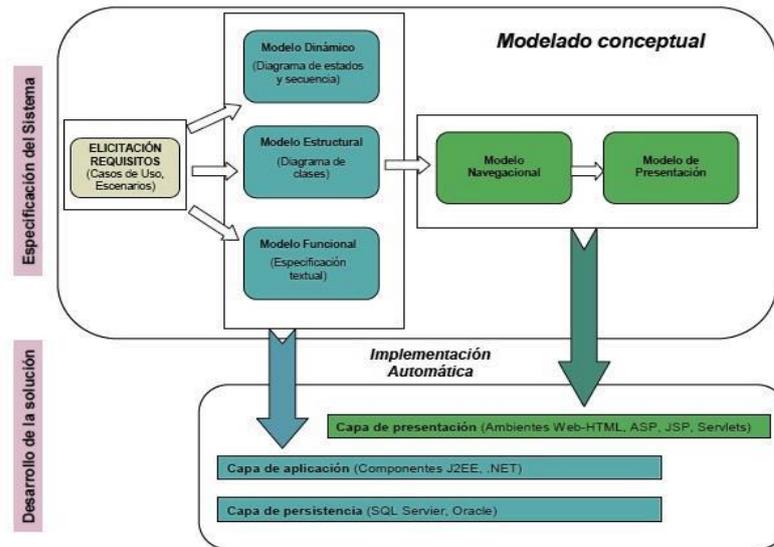


Figura II. 6: Arquitectura Funcional de la Metodología OOWS

Fuente: Tesis, Desarrollo Dirigido por Modelos de Aplicaciones web. (Quintero, 2008)

2.6.1.1. Especificación del Sistema

Esta etapa consta de subprocesos para determinar las especificaciones y el comportamiento que debe ofrecer el sistema para satisfacer los requisitos de usuario identificados.

Captura de requisitos funcionales: que se realiza mediante diagramas de casos de uso y escenarios, que luego son usados para construir el modelo conceptual.

Modelado conceptual clásico: captura la estructura y el comportamiento del sistema desde los puntos de vista: (1) *estructural*, (2) *funcional* y (3) *dinámico*.

(1) *Estructural:* Se establece un *Diagrama de Clases* mediante la definición de la estructura del sistema que se logra estableciendo por un lado: sus clases, operaciones y atributos, y por otro: estableciendo las relaciones entre clases mediante especialización, asociación y agregación.

(2) *Funcional*: captura la semántica de los cambios de estado de los objetos para definir el efecto de los servicios, usando una especificación textual formal.

(3) *Dinámico*: se realizan dos tipos de diagramas (a) el *Diagramas de Transición de Estados*, una descripción de los diferentes ciclos de vida de cada clase del sistema y (b) el *Diagramas de Secuencia* constituido por las interacciones y comunicación entre objetos.

Modelado de la navegación y de la presentación: se establecen los requisitos de navegación por medio de un (1) *Diagrama de Usuarios* y (2) un Modelo Navegacional; y los requisitos de presentación por medio de (3) un Modelo de Presentación, (Pastor, A. & Fons Joan, 2002).

(1) **Diagrama de Usuarios**: se especifican los tipos de usuarios como anónimos, genéricos, registrados, y de acuerdo a ello se establecen los permisos de acceso al sistema.

(2) **Modelo Navegacional**: Aquí se define como se estructura el acceso de los diferentes usuarios que interactúan con el sistema, en función de su objetivo mediante la construcción de diagramas conocido también como grafo navegacional asociado a cada usuario formado por 2 elementos:

Nodos: Son entidades de interacción que facilitan el acceso a datos y funcionalidad más importante para el usuario

Enlaces: Llamados también aristas que relacionan entidades, es decir, proporcionan una unión relación de alcance entre nodos para llegar a cierto objetivo.

Primitivas de Abstracción Básicas

Constituyen los principales componentes para el diseño de una aplicación web como son:

- Mapa Navegacional: “Visión Global de una aplicación web según un perfil de usuario”, es decir para llevar a cabo el diseño de navegación es importante definir el tipo de usuario que va a utilizar el sistema.
- Contexto de Navegación: “Conjuntos de objetos que el usuario irá navegar”, se refiere al diseño de un menú para facilitar al usuario interactuar con los diferentes módulos.

- Vínculo de Navegación: “Indica la relación entre contextos de navegación”, es decir debe existir tipos de aristas que unen a los nodos.
- Clase Navegacional: “Contenido de la información por el cual los usuarios navegarán”, de la correcta definición del contenido depende la funcionalidad del sistema.
- Relaciones: “Maneras de navegar para acceder al contenido de la información”, se refiere a que debe existir diferentes formas de acceso.

(3) **Modelo de Presentación:** Este modelo está directamente relacionado con el modelo de navegación, ya que depende de este para definir la estructura lógica de presentación de los objetos navegacionales, permite la captura de los requisitos básicos que define la presentación de información, orientado a ambientes web.

2.6.1.2. Desarrollo de la Solución.

En esta etapa se propone una estrategia de generación de código basada en componentes para integrar la solución propuesta en ambientes web. En esta etapa se obtendrá una aplicación web, con una funcionalidad equivalente a la especificación inicial según una visión operativa mediante el desarrollo de 3 capas: (Pastor, A. & Fons Joan, 2002)

- a) La primera capa es la de presentación que detalla como la información debe ser presentada a los usuarios, la misma que está conformada por el navegador y el servidor web. El navegador contiene paginas HTML, programas interpretes (JavaScript) y programas compilados (Applets de java), para facilitar la validación de los datos, ayudar en la navegación y animar la interfaz del usuario.
- b) La capa de lógica del negocio, denominada también de aplicación que se encarga por definición de crear la correspondencia entre los datos almacenados y la información que será presentada al usuario a través del servidor Web, es decir, actúa como un middleware para soportar la integración y comunicación entre los componentes, se considera como la parte abstracta entre la interfaz de usuario y la base de datos.

- c) Finalmente se define la capa de datos que se encarga de detallar la representación física de los datos, la misma que está compuesta por los repositorios de datos que pueden ser bases de datos relacionales, orientadas a objetos y/o mixtas de datos XML, es decir son tablas donde se almacenas los datos finales, relacionados entre sí de manera lógica.

2.6.2. CONSTRUCCION DEL MODELO NAVEGACIONAL

Identificación de Agentes: Para iniciar la construcción de este modelo se inicia buscando en el Modelo de Objetos los agentes o entidades principales del sistema y detectar las relaciones entre ellos, el cual involucra lo siguiente:

- Construir los **árboles de agentes**, en el cual debe constar la relación que posee con los demás agentes o entidades.
- Estos árboles están compuestos de: agentes/Clases Base y agentes/SubClases

Construcción de los Mapas: El mapa es un grafo dirigido en el cual los nodos son llamados Contextos Navegacionales y las aristas Enlaces Navegacionales.

Cada Contexto Navegacional es una unidad de interacción que representa una vista cohesiva de datos y funcionalidad (atributos y operaciones de las clases), representada mediante un paquete UML que posee el estereotipo <<Context>>, representados en la figura II.7.

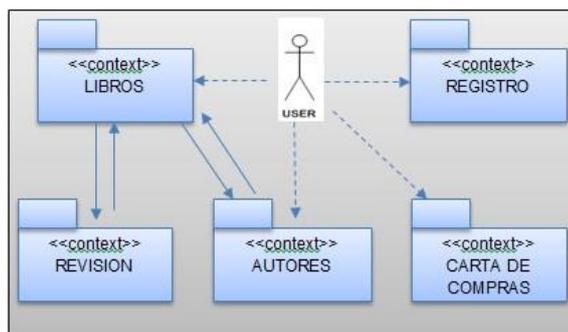


Figura II. 7: Mapa Navegacional

Fuente: Autor

Dependiendo de la forma en la que pueden ser alcanzados los contextos al navegar, se clasifican en uno de dos tipos: si se pueden alcanzar desde cualquier nodo se les llama Contextos de Exploración (indicado con la letra “E”), uno de los cuales se identifica como el Contexto de inicio (Home - indicado con la letra “H”). Si el contexto se alcanza a través de una secuencia de pasos de navegación (Enlaces Navegacionales), es llamado Contexto de Secuencia (indicado con la letra “S”). (Moreno L., 2011)

2.6.3. CONSTRUCCION DEL MODELO DE PRESENTACIÓN

La construcción del siguiente modelo recoge la semántica de presentación de toda la información del sistema y asocia patrones de presentación a los elementos que aparecen en estos nodos navegacionales, este modelo está basado en los elementos definidos en el modelo navegacional.

Los requisitos de presentación de los Contextos Navegacionales se capturan en el Modelo de Presentación como se puede observar en la figuraII.8.

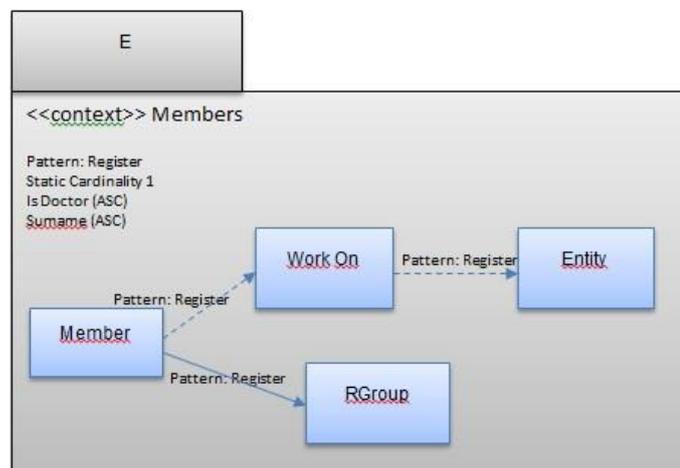


Figura II. 8: Modelo de Presentación

Fuente: Autor

Los contextos tienen un carácter navegacional que permite estructurar la navegación del sistema definiendo la clase y el vínculo navegacional; Clase Navegacional se refiere gráficamente son

clases UML estereotipadas con la palabra reservada <<view>> y el vínculo navegacional define una relación de alcanzabilidad entre contextos de navegación definidos implícitamente.

- **Patrón de Presentación:** Definirá el modo en que la información de las instancias relacionadas será presentada, existen cuatro modos: registro, tabular, maestro-detalle y árbol. Podemos capturar los requisitos básicos para la construcción de interfaces del sistema, a nivel de modelado conceptual, este repositorio de información será utilizado por el generador (compilador) para generar las distintas interfaces para cada usuario.
- **De paginación de la información:** que permite definir la capacidad de desplazamiento vertical (scrolling) sobre la información. Todas las instancias se dividen en bloques lógicos, de tal forma que cada bloque es visible a la vez. Además se ofrecen mecanismos para avanzar y retroceder. El patrón se aplica a la clase directora, a la relación navegacional, a un índice o a un filtro. La información requerida por el patrón incluye: (1) la Cardinalidad, el número de instancias del bloque; (2) el Modo de acceso: secuencial, para acceder al bloque siguiente, previo, primero y último y aleatorio, para acceso directo o circular, para un comportamiento circular del conjunto de bloques.
- **De ordenamiento:** que define un criterio de ordenamiento de la población (con valores ASC=ascendente o DESC=descendente) acorde al valor de uno o más atributos. Se puede aplicar a clases navegacionales, a estructuras de acceso o mecanismos de búsqueda.

Implementación Automática

A partir de la especificación conceptual se construye una solución de software funcionalmente equivalente que está basada en patrones de traducción, los mismos que van desde las primitivas del modelado a las estructuras de implementación.

La implementación automática propone una estrategia de generación de código basada en componentes para integrar la solución propuesta en una arquitectura para la web; además permite obtener una aplicación con funcionalidad equivalente a la especificación inicial según

una visión operativa, dicho proceso se debe realizar en las herramientas adecuadas y cumpliendo los requisitos de la ingeniería inversa.

2.6.4. VENTAJAS Y DESVENTAJAS

Ventajas

- Esta metodología define primitivas navegacionales y de presentación que están integradas en el Modelado Conceptual, que ayudan a una mejor definición en la navegación.
- También contiene una aproximación para definir semántica de navegación en modelos Orientados a Objetos, ya que usa una notación adecuada a la misma.
- Esta metodología es una ampliación de un Método OO de producción de *software* “tradicional” y utiliza la notación UML (adaptada)

Desventajas

- Utiliza etapas extensas de modelamiento ya que contiene contextos y mapas que detallan paso a paso la ubicación de los elementos.

2.7. PRODUCTIVIDAD

La productividad es un instrumento comparativo para gerentes y directores de empresa, ingenieros industriales, economistas y políticos. Compara la producción en diferentes niveles del sistema económico, con los recursos consumidos.

Un error muy común consiste en confundir la productividad con la eficiencia. Eficiencia significa producir bienes de alta calidad en el menor tiempo posible. Por su parte, productividad está cada vez más vinculada con la calidad del producto, de los insumos y del propio proceso.

El mejoramiento de la productividad no consiste únicamente en hacer las cosas mejor; es más importante hacer mejor las cosas correctas. El proceso de producción es un sistema social complejo, adaptable y progresivo. (Pérez, Méndez, Grimán, & Mendoza, s. f.)

La **Productividad del Software** indica que factores como: el personal, la motivación, el trabajo en grupo, la comunicación, el entorno del trabajo, el enfoque del cambio y las herramientas CASE inciden en la productividad del software, (Sommerville I., 2002). Determinada también como una combinación de eficiencia y eficacia, ya que la efectividad está relacionada con el desempeño y la eficiencia con la utilización de los recursos.

La diferencia entre la Eficacia y la Productividad se da a largo plazo ya que la primera de éstas establece un resultado en sólo un período de tiempo y la segunda, necesita los valores de períodos anteriores para establecer si hubo una mejora de la productividad, para ello consideramos factores que afectan a la productividad como son el proceso, producto, y recursos. Existen modelos de Productividad que ayudan al avance del proyecto de software en el desempeño: Objetividad, Generalidad, Significancia, Independencia; así también es importante tomar en cuenta los enfoques de estimación de esfuerzo opinión experta, analogías, descomposición, modelos matemáticos, para una correcta evaluación, (Visconti Z, 2006).

Para mejorar la productividad dentro de un proyecto de software es necesario aplicar conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto con el objetivo de cumplir las necesidades y expectativas, que se definen en tres categorías:

- Alcance, tiempo, coste y calidad.
- Interesados con diferentes necesidades y expectativas.
- Requerimientos identificados (necesidades) y no identificados (expectativas).

Determinar y seleccionar el tipo de personal que interviene tanto en el desarrollo como el uso del software y establecer las relaciones entre las distintas unidades de trabajo, programar cada actividad, evaluar los recursos, determinar las responsabilidades de cada actividad y definir el

uso de los recursos y el presupuesto por período de tiempo (Visconti Z, 2006). Sin embargo para lograr una productividad extensa existen diferentes métricas que mejoran el desarrollo de Software.

- Determinación de tipos de requerimientos
- Métricas del producto y del proceso
- Tamaño y diseño
- Estructuras de datos y de control
- Puntos de función y esfuerzo
- Métricas de calidad y fiabilidad

Según los principios que rigen la ingeniería de métodos (Barrios J., 2002), un método está conformado por un modelo de producto y un modelo de proceso. En el cual define que el modelo de producto es una abstracción de las propiedades comunes que se encuentran en cualquier producto de desarrollo, mientras que el modelo de proceso especifica el conjunto de acciones requeridas para desarrollar un producto de acuerdo al modelo de producto, es decir para comprender el modelo de producto es necesario describir las características del producto y su lógica interna, pero el modelo de proceso se puede ir definiendo de acuerdo al avance del desarrollo, teniendo en cuenta factores como el enfoque, la orientación, la notación.

La selección de indicadores incluye la medición de esfuerzos (en horas-persona) y costes (en dinero) tanto reales como planificados, número de entregables aceptados por el usuario y desviaciones, esfuerzo reutilizado de otros proyectos o inactividad en recursos humanos, número de modificaciones en el producto e información sobre su evaluación (solicitadas, rechazadas y aceptadas), esfuerzo dedicado a la detección y corrección de errores, número de revisiones finalizadas, e información sobre la detección de errores para evaluar la calidad del producto (por ejemplo errores detectados antes y después de la entrega). Asimismo, muchos de

los indicadores incluyen mediciones desglosadas por fase en requerimientos, diseño, codificación y documentación, (Presedo, Aguirregoitia, & Doblado, 2010).

2.8. JSF (JAVA SERVER FACES)

Uno de los tantos frameworks de desarrollo es conocido como JSF el cual está basado en el conocido lenguaje de programación Java. JSF este tiene como fin, simplificar el desarrollo de interfaces web en las aplicaciones JEE(Java Enterprise Edition), (Niño, 2010).

Como características principales de JSF podemos mencionar que este utiliza el patrón MVC (Modelo Vista Controlador) en la creación de las aplicaciones de esta forma:

- JSF Utiliza páginas JSP para generar las vistas de usuario
- Asocia cada vista que se crea a un conjunto de objetos java (clases) las cuales son conocidos como ManagedBeans estos facilitan la forma en que se manipula la información.

JSF también puede ser integrado con otro framework de desarrollo conocido como Hibernate el cual completa el modelo MVC añadiendo una nueva capa, la capa de Persistencia. Normalmente una página web html consta de etiquetas las cuales brindan elementos para la navegación en un portal, estas etiquetas sirven para crear/mostrar botones, textos, imágenes, listas, entre otros. (Moreno L., 2011)

La principal función del controlador JSF es asociar a cada página JSF una clase Java y desde allí manejar la información que el usuario digita. Utilizando JSF le simplifica al programador la forma en que se muestran datos al usuario, leer datos que el usuario escribe, controlar los estados y, otras acciones.

2.8.1. ARQUITECTURA JAVA

La arquitectura J2EE permite desarrollar modelos de aplicaciones multicapa para componentes distribuidos, (Niño, 2010) cualquier número de capas es posible pero comúnmente tenemos:

Cliente: Conformada por la máquina del cliente que soporta navegador web, applets, aplicaciones. J2EE maneja el concepto de clientes “delegados” lo que significa que la interfaz cliente no ejecuta consultas a la base de datos, no implementa lógica de negocio, estas operaciones las realizan tras capas.

Servidor: El servidor de J2EE es el centro de la arquitectura, contiene los componentes web y los objetos que contienen la lógica del negocio. Los componentes web procesan dinámicamente los requerimientos provenientes de los usuarios y construyen respuestas a sus aplicaciones cliente.

Los objetos de negocio implementan tareas dentro del dominio de actividades de la organización. Ambos componentes (lógica de negocio, componentes web) son manejados por el servidor de aplicaciones J2EE, el cual provee importantes servicios como: seguridad, manejo transaccional, de localización de objetos y conectividad remota. Al colocar estos servicios bajo el control del servidor de aplicaciones J2EE, el desarrollador puede enfocarse en la lógica de presentación o negocio.

Servidor de base de datos: Conformado por cualquier motor de base de datos (Sql Server, mySql, PostgreSQL, Oracle, etc..)

CAPÍTULO III

MARCO METODOLÓGICO

En este capítulo se describe detalladamente como y con que métodos se va a realizar la investigación.

3.1. DISEÑO DE LA INVESTIGACIÓN

El método utilizado como guía para la presente investigación es el método Científico, el cual contempla los siguientes puntos:

- El planteamiento del problema que en este caso es el estudio comparativo de las metodologías UWE y OOWS para mejorar la productividad en el desarrollo de una aplicación web.
- Levantamiento de información necesaria.
- Formulación de la Hipótesis.
- Desarrollo de Prototipos
- Análisis e interpretación de Resultados.
- Proceso de Comprobación de la Hipótesis.

El método experimental permite realizar la comparación de las metodologías, mediante el desarrollo de prototipos, que facilita la evaluación de los indicadores.

3.2. TIPO DE ESTUDIO

Para conocer todas y cada una de las características que posee las metodologías UWE y OOWS se realiza una investigación experimental, en la cual ayuda a profundizar los conocimientos de las metodologías a comparar, mediante la utilización de procesos estadísticos que permitan el cálculo del tiempo de desarrollo, la evaluación cualitativa de las métricas del modelo de proceso y producto.

El estudio se efectúa mediante la valoración de los resultados obtenidos, tanto valores numéricos de tiempo como porcentajes de avance en la productividad, el cual se expresa mediante gráficos estadísticos, para luego llevar a cabo el respectivo análisis y la comprobación de hipótesis planteada.

3.3. HIPOTESIS

La aplicación de la metodología OOWS permite una mejor productividad en el desarrollo de una aplicación web.

3.4. DETERMINACIÓN DE INDICADORES Y METRICAS

Los indicadores que se presentan a continuación se basan en los datos del proyecto en estudio y en el registro histórico de proyectos pasados. Estos indicadores están propuestos en base a seis parámetros:

Tabla III. I: Parámetros indicadores y métricas de evaluación

INDICADORES	PARAMETROS	METRICAS
Tipos de Requisitos	Requisitos de Datos, Interfaz de Usuario, Navegacionales,	SI
	Personalización,	ó NO

	Transaccionales y No funcionales	
Propuesta Funcional	Notación del Dominio Notación de Navegación Notación de Presentación	Propia Otra
Diseño de Metodología	Tiempo de diseño	Promedio de tiempo del diseño de cada diagrama.
		Promedio de tiempo de diseño de la metodología completa.
Modelo de Producto	Modelo del producto	Implícito Explicito No existe
	Notación del producto	Formal (fórmulas matemáticas) Semi-formal (Cuadros, gráficos) Informal (Textual)
	Orientación del modelo	Orientado a objetos Orientado a procesos Basado en ejemplos Indefinidos
	Perspectiva del producto	Funcional Tecnológica Estructural
Modelo de Proceso	Orientación	Orientado al producto Orientado a la actividad Orientado a la decisión (sentencias)
	Enfoque de programación	Preventivo (cascada) Evolutivo (adaptativo)
	Claridad	Correcta definición de la estructura del proceso
	Estructura del proceso	Fases/tareas
	Modo de ejecución del proceso	Interactivo Secuencial
	Cantidad de software desarrollado por unidad de	<ul style="list-style-type: none"> Número de puntos de función desarrollados.

Puntos de Función	tiempo de trabajo.	<ul style="list-style-type: none">• Tiempo empleado
	Ritmo de entrega o Capacidad de desarrollo por unidad de tiempo transcurrido.	<ul style="list-style-type: none">• Número de puntos de función desarrollados.• Tiempo transcurrido
	Funcionalidad entregada por unidad de coste.	<ul style="list-style-type: none">• Número de puntos de función desarrollados.• Coste total

Fuente: Autor

3.4.1. INDICADORES

A continuación se define cada uno de los parámetros determinados anteriormente.

Parametros del indicador Tipos de Requerimientos

De acuerdo a la ingeniería de requisitos se conoce que existe varios tipos de requerimientos pero para el desarrollo de aplicaciones web se toma en cuenta los principales que son: Requisitos de Datos, Interfaz de Usuario, Navegacionales, Personalización, Transaccionales y No funcionales, los cuales serán determinados cuántos y cuáles de ellos cumplen las metodologías en estudio.

Parametros del indicador Propuesta Funcional

Para este indicador es fundamental medir la Notación del Dominio, Notación de Navegación, y Notación de Presentación, es decir, conocer si por cada metodología usan diferentes estándares o basado en UML.

Parametros del indicador Diseño de Metodología

Tiempo de diseño: Este indicador está estipulado para medir el tiempo que el desarrollador se demora en diseñar cada uno de los diagramas.

Luego de contabilizar el total de diagramas de cada metodología, obtener un promedio de tiempo total.

Parametros del indicador Modelo de Producto

Modelo del producto: Involucra las características que evalúan como esta expresado el contenido de la aplicación.

Notación del producto: Se refiere a los componentes utilizados en el diseño de los diferentes diagramas y modelado como el uso de fórmulas matemáticas, gráficos, cuadros y textos.

Orientación del modelo: Se expresa que durante el desarrollo de la aplicación el modelo debe definir a donde está orientada la programación.

Perspectiva del producto: El producto final tiene como objetivo ser útil en algún campo de trabajo, por ello es necesario establecer el tipo de perspectiva que tiene cuando ha sido desarrollado como puede ser funcional, tecnológico o estructural.

(Barrios & Mendoza, 2004)

Parametros del indicador Modelo de Proceso

Orientación: Se refiere a que tipo de orientación posee durante el proceso, ya que involucra el producto, actividad, o decisión.

Enfoque: Durante el proceso de desarrollo se evalúa los estados y actividades realizadas, las cuales se basan en un desarrollo preventivo o evolutivo.

Claridad: Se refiere a la precisión y facilidad de interactuar con los componentes estructurales que utiliza durante el proceso de desarrollo.

Estructura del proceso: Son las fases o pasos que posee la metodología para el diseño y desarrollo de la aplicación.

Modo de ejecución del proceso: Está catalogado como el tipo de ejecución que se lleva a cabo secuencialmente o por iteración.

(Barrios & Mendoza, 2004)

Parametros del indicador Puntos de Funcion

También se menciona como parámetro el cálculo de líneas de código, considerado como factor importante en la evaluación de la productividad de una aplicación web. Estableciendo como indicadores los siguientes:

Cantidad de software desarrollado por unidad de tiempo de trabajo: Se calcula dividiendo la medición del software desarrollado o entregado de conformidad, por el tiempo empleado.

Ritmo de entrega o Capacidad de desarrollo por unidad de tiempo transcurrido: Un elemento importante para competir hoy en día es el tiempo de entrega del producto (bien o servicio) al cliente, por lo que son de interés las métricas del tiempo relacionado con la capacidad de desarrollo por unidad de tiempo.

Funcionalidad entregada por unidad de coste: El término coste es utilizado en la bibliografía de métricas de software con significados distintos: a) cantidad de recursos en unidades físicas que se correspondería más con el esfuerzo o consumo de recursos necesarios, b) valor monetario de los recursos, que se corresponde con el concepto contable.

3.4.2. MÉTRICAS

Las métricas establecidas para el modelo del producto y el modelo del proceso son evaluadas cualitativamente, ya que de acuerdo al indicador que corresponde se expresa gráficamente los valores obtenidos. Mientras que para el parámetro de líneas de código se detalla las métricas con sus respectivas formulas (Garmus, 2001).

- Número de puntos de función desarrollados y Tiempo empleado: **Productividad TI** =
Total PF Completos / Total de esfuerzo en TI

- Número de puntos de función desarrollados y Tiempo transcurrido: **Ritmo de entrega** = $\#PuntoFunción\ Completos / Tiempo\ transcurrido$
- Número de puntos de función desarrollados y Coste total: **Funcionalidad-coste** = $\#PuntosFunción\ Completos\ desarrollados / Coste\ total$.
- **Promedio de tiempo del diseño de cada diagrama:** Se toma como muestra el tiempo empleado en horas de 5 diagramas, para posteriormente, hallar un promedio del tiempo empleado.
- **Promedio de tiempo de diseño de la metodología completa:** Suma total del número de diagramas desarrollados por cada metodología, multiplicado por el tiempo obtenido en el ítem anterior.
- **Escala de valoración:** Formada por una tabla de acuerdo al número de ítems a ser valorados el cual requiere un valor por nominación de mayor a menor.

Criterios para propuestas funcionales

- Notación del dominio: es necesario que la metodología cuente con herramientas para poder modelar los elementos propios del dominio de la aplicación.
- Notación de navegación: debe ser posible representar la composición lógica y las estructuras de navegación de las páginas web.
- Notación de presentación: para poder cumplir con los aspectos generales de las propuestas funcionales se requieren componentes que permitan modelar la representación gráfica que tomará el hipertexto.

3.5. MÉTODO DE COMPARACIÓN DE METODOLOGÍAS

La comparación se efectuara en base a seis indicadores específicos definidos anteriormente que serán tomados en cuenta durante el proceso de desarrollo, ya que es difícil imaginar un proyecto en el que no sea importante algún nivel de medición y control sobre cada uno de los siguientes *atributos* para la consecución de un resultado exitoso:

- Características del producto (requerimientos implementados y en funcionamiento),
- Notación y diseño (herramientas, formato de diagramas, tiempo de demora)
- Esfuerzo (tiempo gastado durante las actividades del trabajo),
- Programa (logro de los hitos objetivamente medidos),
- Coste (gastos en las distintas clases de recursos, incluidos los recursos humanos),
- Progreso (productos completados, aceptados y valores de referencia),

La comparación para el indicador 1 consiste en verificar los tipos de requisitos que determina en cada uno de los prototipos y hallar el valor porcentual de acuerdo al número de ítems cumplidos.

Para el indicador 2 se propone una tarea más sencilla ya que sus ítems son tres opciones, el cual se califica en base a tabla de valoración con escala de 1 a 3, y luego procede a determinar los porcentajes.

El indicador 3 hace referencia al diseño de los diferentes diagramas para lo cual se toma el tiempo exacto en horas que demora, luego se halla el promedio por diagrama para calcular el tiempo total de diseño por cada prototipo.

Para los indicadores 4 y 5 se establece una serie de métricas que van a ser evaluadas por medio de una tabla de valoración para luego hallar la sumatoria y promedio por cada prototipo.

El último indicador se encarga de la estimación de los puntos de función para lo cual primero toma como referencia a una estimación previa en el software de COCOMO, luego realiza la toma de los datos de desarrollo de líneas de código por cada prototipo para llegar a una conclusión.

3.6. TÉCNICAS Y HERRAMIENTAS UTILIZADOS PARA LA MEDICIÓN

3.6.1. TÉCNICA DE MEDICIÓN

Escala de Likert: Es una escala psicométrica comúnmente utilizada en cuestionarios y es la escala de uso para la investigación. Al responder a una pregunta de un cuestionario elaborado con la técnica de Likert, se especifica el nivel de acuerdo o desacuerdo con una declaración (elemento, ítem o reactivo o pregunta). (Franco, s. f.)

Elaboración de la escala

1. Preparación de los ítems iniciales; se elaboran una serie de enunciados afirmativos y negativos sobre el tema o actitud que se pretende medir, el número de enunciados elaborados debe ser mayor al número final de enunciados incluidos en la versión final.
2. Asignación de puntajes a los ítems; se le asigna un puntaje a cada ítem, a fin de clasificarlos según reflejen actitudes positivas o negativas.
3. Asignación de puntuaciones a los sujetos; la puntuación de cada sujeto se obtiene mediante la suma de las puntuaciones de los distintos ítems.
4. Análisis y selección de los ítems; mediante la aplicación de pruebas estadísticas se seleccionan los datos ajustados al momento de efectuar la discriminación de la actitud en cuestión, y se rechazan los que no cumplan con este requisito.

Estimación de puntos de función

Para efectuar una correcta medición es necesario seguir un proceso ordenado, para ello se utiliza una metodología de estimación que se detalla de la siguiente manera:

Entrada: Definición del problema

Paso 1: Estimar tamaño (PF)

Paso 2: Transformar tamaño (PF a KLOC)

Paso 3: Estimar tiempo y esfuerzo (COCOMO)

Paso 4: Distribuir tiempo y esfuerzo (COCOMO)

Paso 5: Normalizar a tiempo calendario (ESTERLING)

Salida: esfuerzo, costos, producción

3.6.2. HERRAMIENTA DE MEDICIÓN

COCOMO: El **Modelo Constructivo de Costos (CO**nstructive **CO**st **MO**del) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente.

Modelos de estimación: La ecuaciones que se utilizan en los tres modelos son:

- $E = a(Kl)^b * m(X)$, en persona-mes
- $Tdev = c(E)^d$, en meses
- $P = E/Tdev$, en personas

dónde:

- E es el esfuerzo requerido por el proyecto, en persona-mes
- $Tdev$ es el tiempo requerido por el proyecto, en meses
- P es el número de personas requerido por el proyecto
- a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo
- Kl es la cantidad de líneas de código, en miles.
- $m(X)$ Es un multiplicador que depende de 15 atributos.

3.7. TÉCNICAS Y FUENTES DE RECOPIACIÓN DE INFORMACIÓN

Para la recopilación de la información necesaria que sustente el presente trabajo de investigación, se ha establecido como técnicas las siguientes:

- Revisión bibliográfica de información sobre ingeniería de software, aplicaciones y metodologías web.
- Observación y control de tiempo durante el desarrollo de los prototipos.
- Técnicas de Comprobación de hipótesis.
- Pruebas.

Como fuentes principales de información se ha establecido utilizar las siguientes:

Fuentes Primarias: Google Academico (libros revistas científicas, tesis, conferencias), Libros de Ingeniería de Software, Observacion en el desarrollo de prototipos.

Secundarias: Textos, Revistas, Documentos, Papers.

TÉCNICAS ESTADÍSTICAS

Para determinar una muestra representativa se utiliza el muestreo probabilístico o Aleatorio por ejecutar una selección sistemática

Técnicas para comprobar la hipótesis

- Incremento porcentual
- Sumatoria de pesos
- Distribución T Student

De las técnicas que se menciona se usará Distribución T Student para la comprobación de hipótesis, mediante el calculo del intervalo de confianza del 95%.

3.8. PROCESAMIENTO DE LA INFORMACIÓN

En esta etapa se considera importante como va a ser procesada la información recolectada, por medio de la descripción de los pasos realizados durante el procesamiento de la información.

3.8.1. CONSTRUCCIÓN DE PROTOTIPOS

Con la información adquirida durante la investigación sobre las metodologías a comparar se procede a la construcción de los prototipos, es decir uno por UWE y otro por OOWS del módulo Ingreso de Matricula, el mismo que ha sido elegido por contener las características pertinente para la investigación, este módulo se realaciona con las tablas: inscripción, alumno, padres, representante, persona, matricula, curso y paralelo, considerando dentro del marco de comparación los indicadores determinados anteriormente.

Estos prototipos se desarrollan en la misma herramienta pero siguiendo su proceso independiente; iniciando por el diseño de la arquitectura de acuerdo a los diagramas, obtención de requerimientos, modelado de la base de datos y finalmente la implementación.

3.8.2. REALIZACIÓN DE LA COMPARACIÓN

Luego de haber construido los prototipos con cada una de las metodologías propuestas se lleva a cabo la comparación, donde se plantea una tabla de valores por cada indicador, que facilitará la evaluación de acuerdo al cumplimiento de los parámetros, posteriormente se realiza la sumatoria y se muestra la grafica de resultados.

3.8.3. OBTENCIÓN DE RESULTADOS

La obtención de resultados se realizará con la sumatoria de los valores dados en la calificación de las metodologías por cada indicador, y la representación grafica de dichos resultados se mostrará en barras.

3.8.4. ANÁLISIS DE RESULTADOS

De acuerdo a los resultados obtenidos se lleva a cabo un análisis profundo entre los valores obtenidos estadísticamente de la comparación realizada anteriormente, en el cual se elige a la metodología adecuada que obtuvo los mejores resultados; tomando en cuenta los pesos obtenidos para cada metodología en una sumatoria final y para su comprobación se aplicará el modelo matemático de T Student en el cual se verifica los índices de confianza de los resultados obtenidos.

CAPÍTULO IV

ANÁLISIS COMPARATIVO DE LAS METODOLOGÍAS WEB UWE Y OOWS.

En este capítulo se desarrolla los prototipos y posteriormente la comparación de las metodologías, mediante la evaluación de los indicadores.

4.1. DESARROLLO DE LOS PROTOTIPOS

En esta etapa se desarrolla los respectivos prototipos del módulo “Matricula” .

4.1.1. PROTOTIPO DE LA METODOLOGÍA UWE

La estrategia de diseño UWE se basa en modelos que se construyen durante la fase de análisis, principalmente el modelo conceptual y el modelo de procesos. UWE introduce clases específicas de procesos como parte de un modelo separado, que ofrece una interfaz al modelo de navegación. A continuación se realiza el desarrollo de prototipo para el modulo Matrícula.

4.1.1.1. Fase de Análisis de Requisitos

Describe el proceso que el administrador realiza al efectuar la matrícula de los alumnos, como se observa en la figura IV.9, el cual facilita la determinación de requerimientos.

El administrador accede al sistema y selecciona el menu matricula, luego busca la inscripción previa para continuar la matricula, ingresa los datos del alumno, padres, representante, curso.

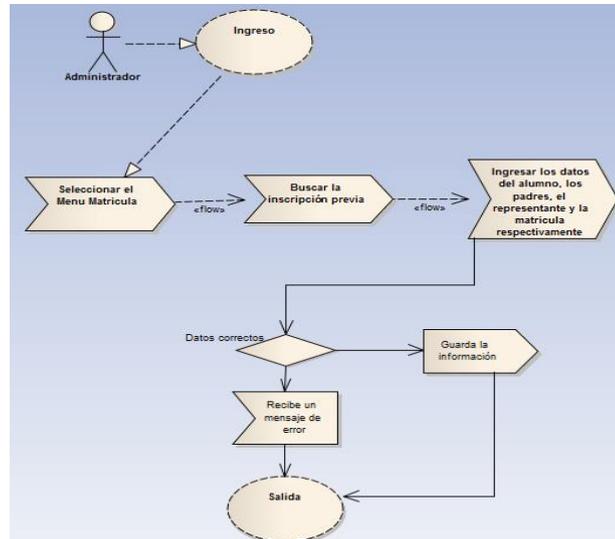


Figura IV. 9: Diagrama de Análisis de Requisitos

Fuente: Autor

Diagrama de Casos de uso: Los siguiente diagramas permiten especificar paso a paso lo que lleva a cabo el administrador cuando ingresa una matrícula, seleccionando el módulo de matrícula y llenando todos los datos, es decir el funcionamiento.

Como se observa en la figura IV.10, el administrador debe seleccionar el modulo matricula, luego agregar nueva, y busca una inscripción previa como acción que precede a llenar datos.

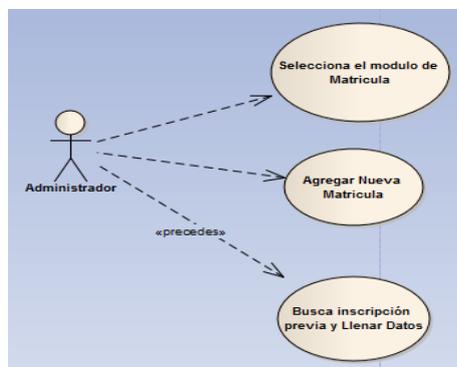


Figura IV. 10: Diagrama de Usuario de ingreso a modulo

Fuente: Autor

Una vez encontrada la inscripción el administrador, puede llenar los datos de padres seleccionando la opción respectiva, figura IV.11.

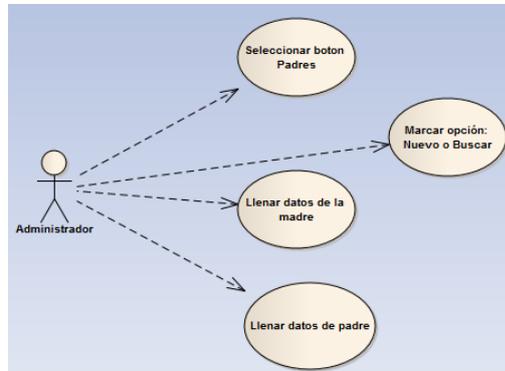


Figura IV. 11: Diagrama de Usuario de ingreso de datos de padres

Fuente : Autor

Para el ingreso de datos del representante el administrador marca la opción requerida de (Padre) (Madre) u (Otros), (figura IV.12) y luego llenar datos si es necesario.

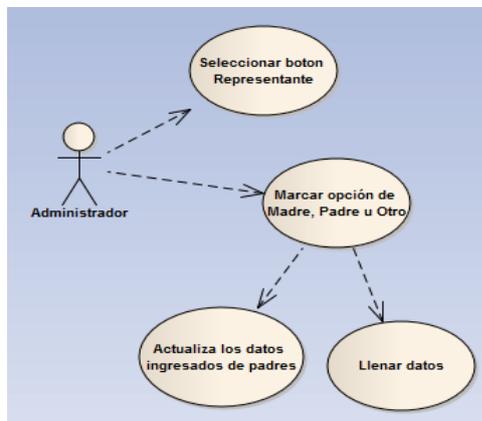


Figura IV. 12: Diagrama de Usuario de Ingreso de datos de Representante

Fuente: Autor

4.1.1.2. Fase de Diseño Conceptual

Se desarrolla el diseño de los diagramas de componentes y objetos que ayudan a identificar la arquitectura del sistema, como están compuestos y las identidades que interactúan durante el proceso de matrícula como se indica en las figuras respectivamente.

Diagrama de componentes

Describe como está compuesta la arquitectura del módulo a ser desarrollado, en este caso el administrador ejecuta la aplicación (módulo matrícula) que está conectada a la base de datos, el cual interactúa con la interfaz como se observa en la figura IV.13, la misma que contiene interfaces tanto para el ingreso al modulo como para llenar datos de padres y de representante.

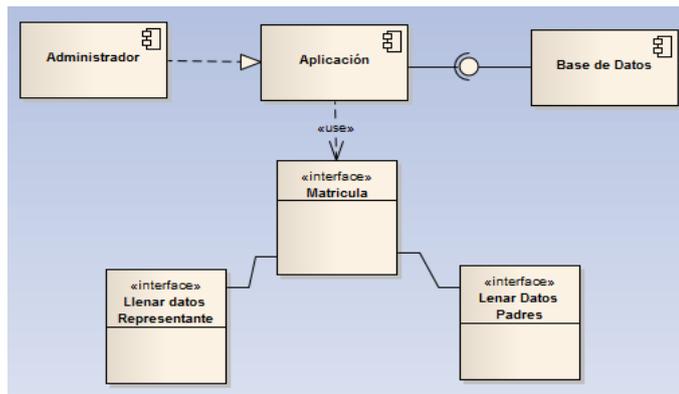


Figura IV. 13: Diagrama de Componentes

Fuente: Autor

Diagrama de objetos

Se define los objetos involucrados en el módulo matrícula como se puede observar en la figura IV.14, objetos Inscripción, Matrícula, Alumno, Representante, Padres, Persona, y Curso.

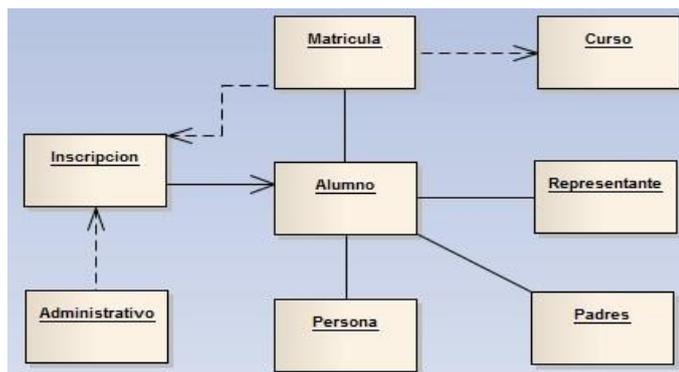


Figura IV. 14: Diagrama de Objetos

Fuente: Autor

Posteriormente se realiza el refinamiento del modelo conceptual, agregando atributos y métodos a las clases identificadas (figura IV.15).

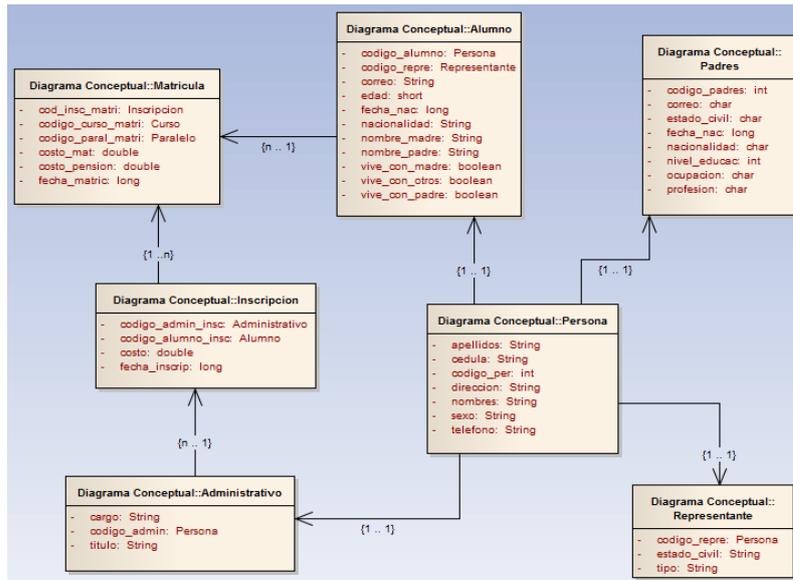


Figura IV. 15: Refinamiento del Modelo Conceptual

Fuente: Autor

4.1.1.3. Fase de Diseño Navegacional

Constan los nodos o clases que involucra la matrícula de un alumno, indicando la navegación que puede realizar el usuario al momento de interactuar.

Modelo navegacional, nodos y enlaces

De acuerdo al modelo de procesos para un correcto diseño del modelo navegacionales indispensable diseñar tres vistas o diagramas que son: una vista de integración con el modelo navegacional, vista estructural y vista de flujos de procesos.

En la figura IV.16, se muestra que en la vista de integración con el modelo navegacional lideran el nodo Navegacional vinculado con el nodo Link, de donde parten las clases navegacionales que permiten la facilidad de desplazarse por el menú.

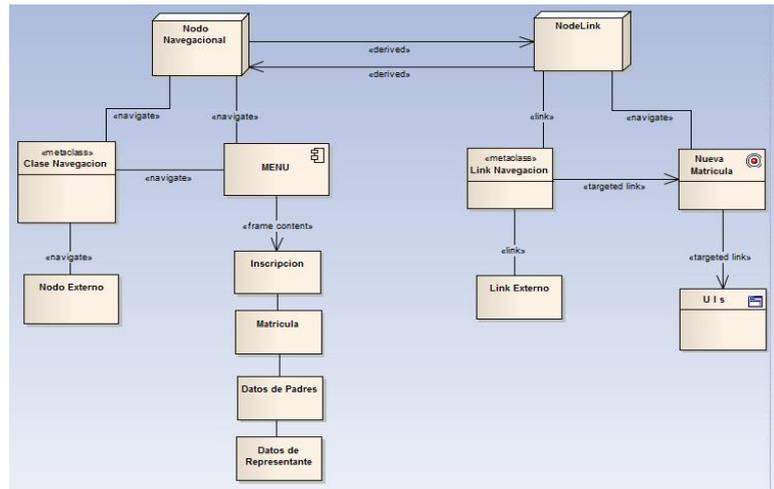


Figura IV. 16: Vista de Integración con el modelo de Navegación

Fuente: Autor

Vista Estructural: Está compuesta por los nodos de navegación que involucran el proceso del módulo matricula (figura IV.17), representados como procesos y enlazados por las clases Link y Navigate.

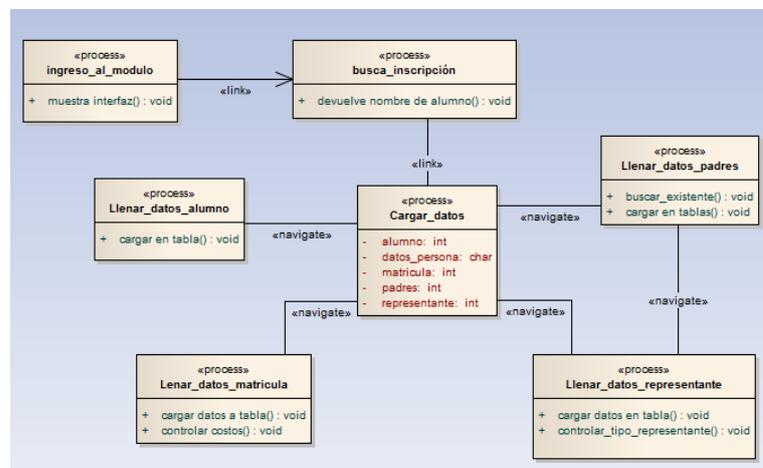


Figura IV. 17: Vista Estructural del Modelo Navegacional

Fuente: Autor

Vista de comportamiento o modelo de flujo del proceso: Cada proceso del módulo matricula tiene asociado un <Modelo de flujo de procesos>, expresado como un Diagrama de Actividad UML. Es decir, la Figura IV.18 muestra el Modelo de flujo de procesos para el módulo de matrícula, en el cual cada actividad es un estado de sub-actividad o un estado de llamada (call).

Los flujos de objeto (objectflow) del diagrama, modelan las entradas y salidas del usuario. Los estados de llamada pueden estereotiparse con la palabra clave <<transaccional>> para expresar el carácter transaccional de este tipo de estados.

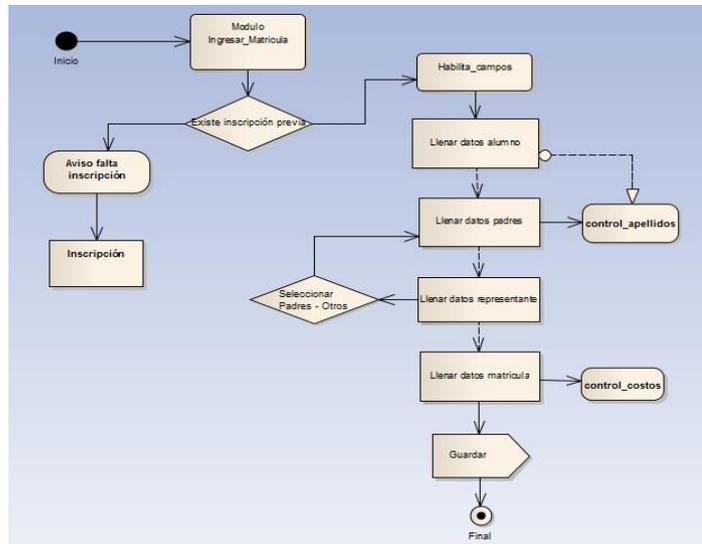


Figura IV. 18: Vista de Comportamiento

Fuente: Autor

4.1.1.4. Fase de Presentación

Contiene la lógica de interfaz de usuario, mediante el proceso de modelar la clase abstracta definido en el diagrama de navegación, permite derivar las operaciones de navegación, entidades y enlaces contenidos (figura IV.19).

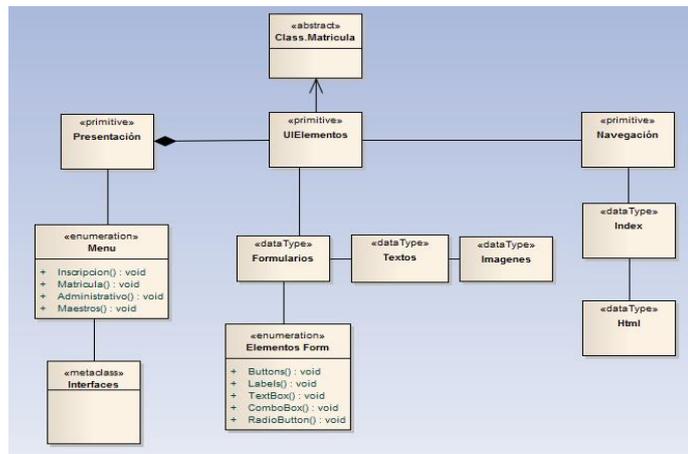


Figura IV. 19: Modelo de Presentación

Fuente: Autor

4.1.1.5. Modelos complementarios

Modelo de Usuario: El diseño de diagramas de casos de uso en esta etapa es muy importante para determinar los tipos de usuarios que hacen uso del sistema, que para este caso que se diseña solo un módulo se utiliza el usuario Secretaria quien va a desempeñar el papel de Administrador como se observa en la figura IV.20.

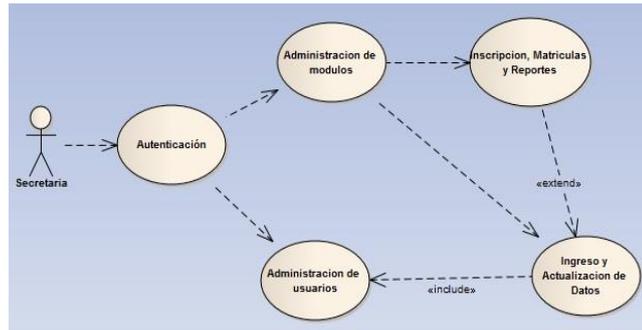


Figura IV. 20: Diagrama del Modelo de Usuarios

Fuente: Autor

Modelo de Adaptación: Se realiza el refinamiento del modelo conceptual y de navegación, agregando métodos a las clases identificadas, y la acción que cumple cada operación mediante el uso de reglas y triggers, para ello se diseña el diagrama de Secuencia (figura IV.21), donde el administrador ingresa a matricula, busca los datos en inscripción, llena los datos de alumnos, posteriormente de los padres, marca la opción del representante y guarda en la base de datos.

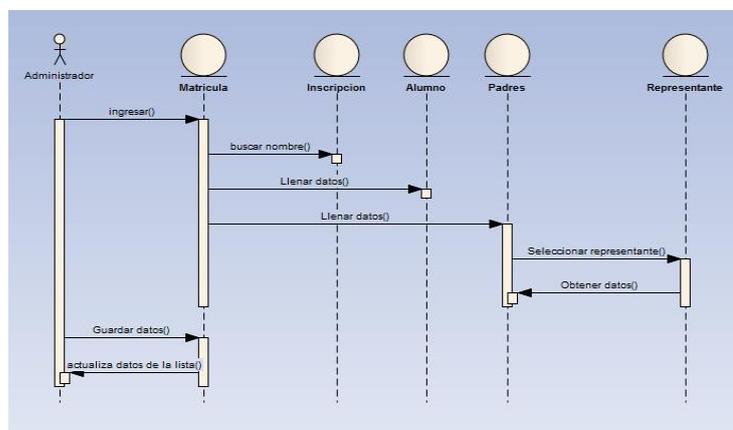


Figura IV. 21: Diagrama del Modelo de Adaptación

Fuente: Autor

Modelo de Tareas: Se considera la definición de casos de usos del sistema por estar considerados como tareas en nivel de análisis, para lograr un correcto diseño del diagrama de actividades (figura IV.22) el cual nos permite identificar todos y cada uno de los procesos.

Se utiliza los objetos creados en el modelo conceptual, Matricula, Inscripción, Alumno, Padre, Representante, de manera que indiquen el procedimiento correcto dentro del módulo de matrícula tomando al diagrama de actividades como "mapas de carreteras" del sistema de comportamiento funcional.

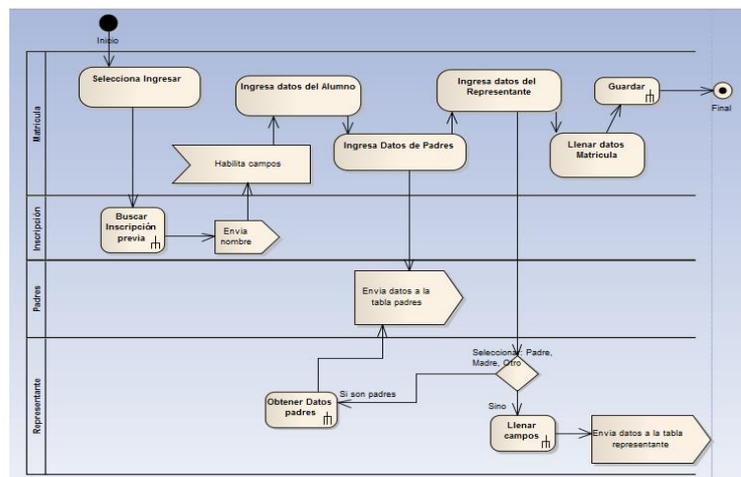


Figura IV. 22: Diagrama de Actividades

Fuente: Autor

4.1.1.6. Base de Datos

El modelado de la base de datos es la parte más importante de esta etapa, ya que es donde se almacena la información del ingreso de matrícula, para ello se utiliza la herramienta DBMS PostgreSQL 9.2.

En la figura IV.23 se puede observar el modelado de la base de datos con las tablas involucradas en el módulo de Ingreso de matrícula.

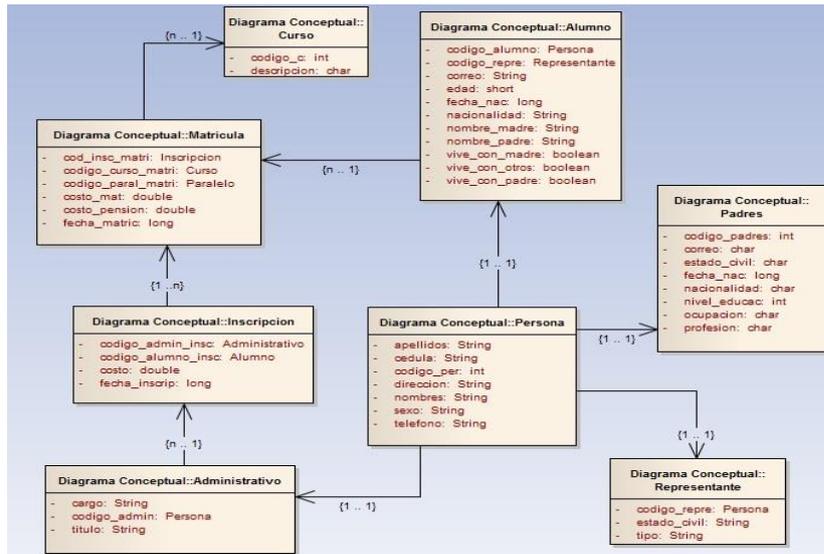


Figura IV. 23: Diagrama de Base de Datos

Fuente: Autor

4.1.1.7. Implementación o programación de la aplicación

La siguiente fase que se describe a continuación es el desarrollo o codificación del módulo de Matricula. S inicia por la creación de un nuevo proyecto, configuración de las librerías, el desarrollo de clases, funciones, controladores, y vistas en HTML, figura IV.24.

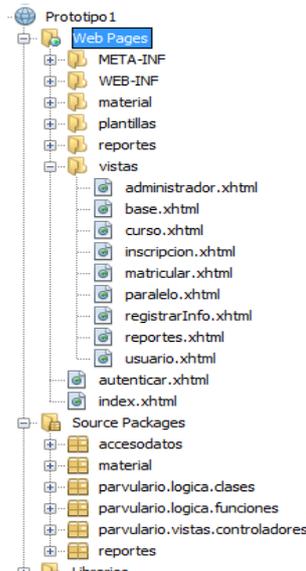


Figura IV. 24: Implementación del Prototipo UWE

Fuente: Autor

De acuerdo a la programación de las vistas en lenguaje HTML, se ejecuta y muestra en el navegador (figura IV.25) la pantalla de ingreso al módulo de matrícula, en el cual consta el panel de navegación hacia otros módulos, datos de selección y botones correspondientes.



Figura IV. 25: Pantalla de ingreso al módulo

Fuente: Autor

Finalmente al dar clic en el botón Ingresar se muestra la pantalla de ingreso de datos de las tablas que interactuar en el módulo matrícula como se indica en la figura IV.26.



Figura IV. 26: Pantalla de ingreso de datos

Fuente: Autor

4.1.2. PROTOTIPO DE LA METODOLOGÍA OOWS

El siguiente prototipo se desarrolla de acuerdo a las etapas mencionadas anteriormente, siguiendo el patrón de diseño conceptual y navegacional para concluir en la implementación.

4.1.2.1. Especificación del Sistema

Determinación de Requisitos: Para la determinación de requisitos se desarrolla el diseño del diagrama de casos de usos (figura IV.27) para posteriormente realizar la conceptualización.

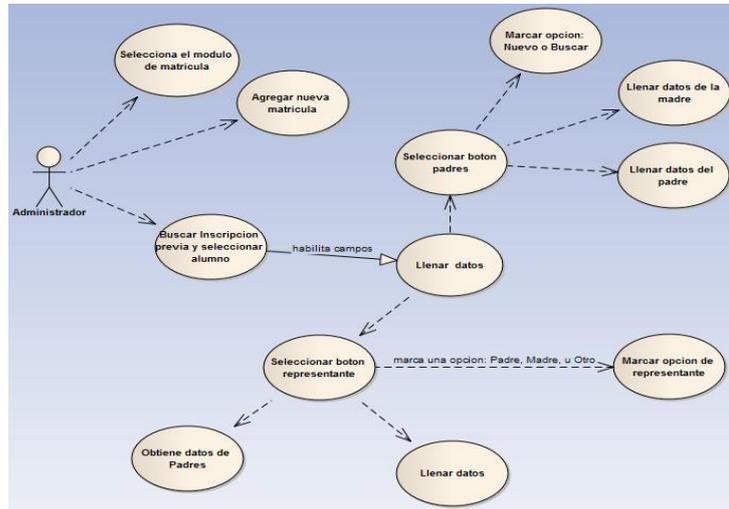


Figura IV. 27: Diagrama de Caso de uso del Prototipo2

Fuente: Autor

Listado de Requisitos: Para este módulo se especifica los siguientes requerimientos que incluye los demás tipos de requerimientos incluyendo los funcionales.

- Realizar la inscripción de alumnos
- Matricular alumnos
- Registro de alumnos
- Registro de Padres de familia
- Registro de representante

Modelado Conceptual

Se realiza la captura de la estructura y el comportamiento del sistema desde los 3 puntos de vista fundamentales.

Vista Estructural: Se desarrolla el diseño del diagrama de clase definiendo cada uno de los objetos con sus respectivos atributos, operaciones y relaciones, figura IV.28.

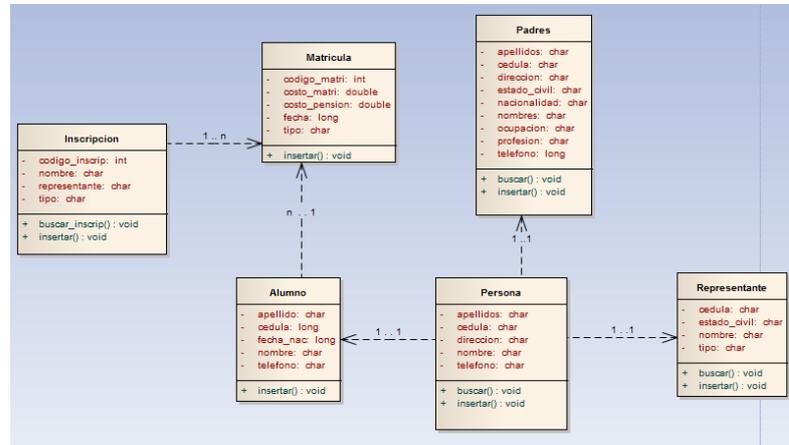


Figura IV. 28: Vista estructural del modelo conceptual

Fuente: Autor

Vista Funcional: El esquema funcional está definido de la siguiente manera; el administrador inicia el proceso con la selección del módulo matricula, en el cual ingresa en estado nuevo para proceder a ingresar datos, luego el sistema solicita la selección de la inscripción previa del alumno que lleva a un estado de espera mientras realiza la consulta, posteriormente a ser encontrado el nombre del alumno entra en estado de listo para llenar campos del alumno, padres, representante y matrícula.

También es necesario especificar que la llenar los datos de los padres llega al estado de búsqueda cuando solicita seleccionar su existencia o no, mientras que el representante pueden ser los mismos padres u otras personas, el cual obliga al administrador a entrar en un estado de búsqueda, espera y actualización.

Una vez terminado el llenado de datos, al seleccionar el botón guardar entra al estado de listo y muestra la nueva matrícula en la lista externa.

Vista Dinámica: Para el modelado conceptual desde esta vista es fundamental el desarrollo de diagramas de Transición de Estados y diagramas de Secuencia.

El diagrama de transición de estados (figura IV.29), como su nombre lo indica describe los estado por los cuales va transcurriendo los procesos desde que el administrador inicia la aplicación. Inicia en un estado Nuevo, luego pasa a, estado Buscando y Espera, finalmente un estado guardado.

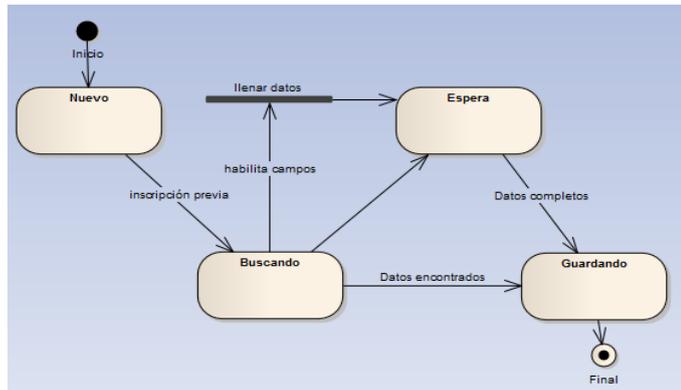


Figura IV. 29: Diagrama de Transición de Estados

Fuente: Autor

Es indispensable conocer el proceso que realiza el usuario al ejecutar el modulo desarrollado por esto se diseña el diagrama de secuencia (figura IV.30), que consta de las entidades involucradas Matricula, Inscripcion, Alumno, Padres, y Representante, el cual realiza las operaciones repectivas.

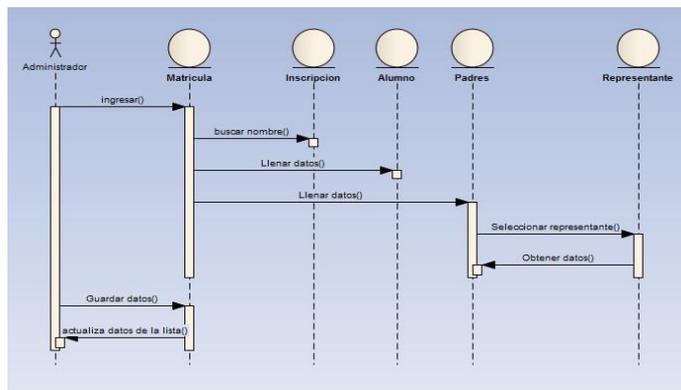


Figura IV. 30: Diagrama de Secuencia

Fuente: Autor

Modelado de Navegación y Presentación

El siguiente modelado permite establecer los requisitos de navegación por medio de un Diagrama de Usuarios (figura IV.31) y un Modelo Navegacional. Mientras que para determinar los requisitos de presentación se desarrolla un Modelo de Presentación.

Diagrama de Usuarios

Para identificar el rol de usuario se diseña el diagrama de caso de uso en el que se identifica las acciones que involucra el desarrollo del módulo matricula, para ello solo se idéntica un solo usuario que es el Administrador, el cual dentro del parvulario cumple con ese rol la secretaria.

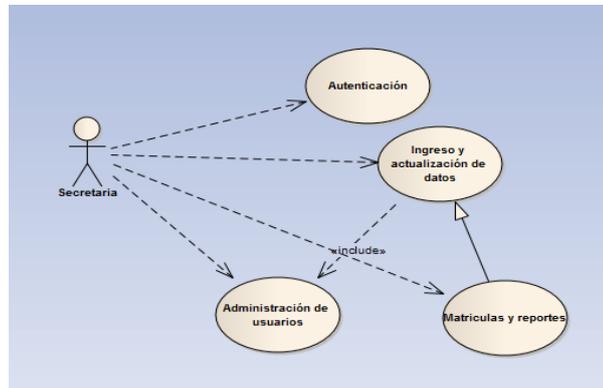


Figura IV. 31: Diagrama de usuario secretaria

Fuente: Autor

Modelo Navegacional

El modelado parte del diseño de los nodos y enlaces de la navegación para concluir en el desarrollo de contexto el mismo que permite el diseño de la presentación. La figura IV.32 que se muestra a continuación describe el diseño de la navegación que posee el prototipo, incluyendo los nodos de navegación y sus respectivos enlaces.

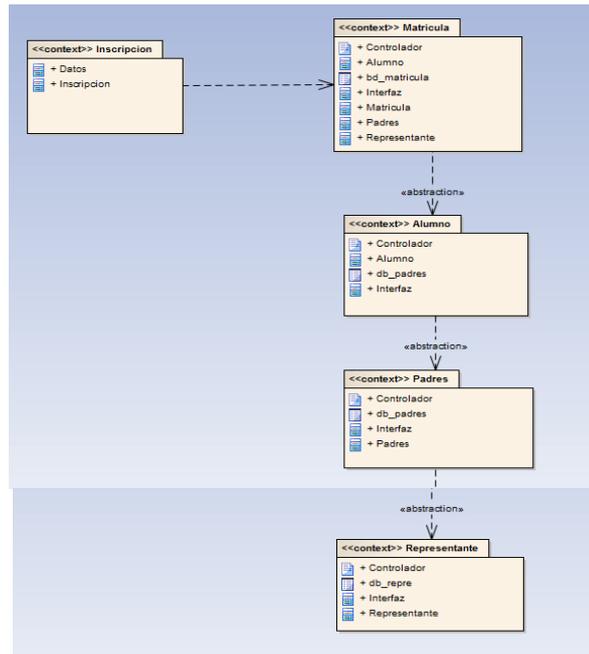


Figura IV. 32: Diagrama de Navegación

Fuente: Autor

Modelo de Presentación

El diseño del modelo de presentación realiza la definición detallada de cada nodo tomando a dichas entidades como contextos en el cual las propiedades patrón de presentación de los descriptores de las acciones en cada clase se utilizan para la construcción del Modelo de Presentación de la aplicación Web. A cada una de las IU que se han definido en el Modelo de Navegación se les aplica el patrón definido en el descriptor de la acción Web que le dio origen.

Contextos de Presentación: La definición de contextos involucra de donde parte cada entidad, el patrón y los descriptores de acción (figuras IV.33 - IV.37).

Contexto Inscripción: muestra los elementos que contiene la vista, los mimos que serán almacenados en la base de datos con los siguientes descriptores de acción: patron Tabular e información Descendente.

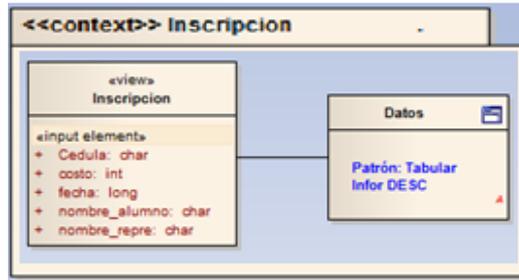


Figura IV. 33: Contexto de Inscripción

Fuente: Autor

Contexto Matricula: muestra la base de datos, el controlador, los elementos que contiene la vista y, las tablas relacionadas a la misma; con los siguientes descriptores de acción: patron Registro, cardinalidad de 1 . . n, e información Ascendente.

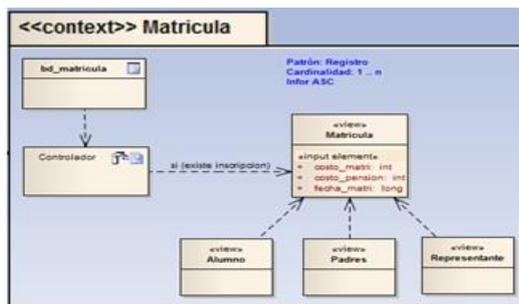


Figura IV. 34: Contexto de Matrícula

Fuente: Autor

Contexto Alumno: muestra la base de datos, el controlador y, los elementos que contiene la vista; con los siguientes descriptores de acción: patron Registro, cardinalidad de 1..1, e información Ascendente.

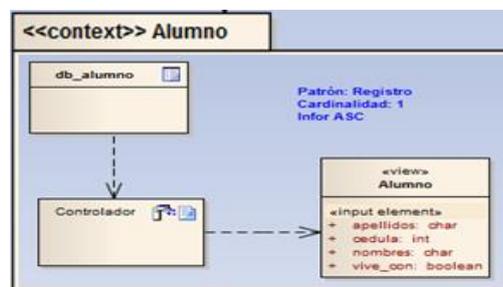


Figura IV. 35: Contexto de Alumno

Fuente: Autor

Contexto Representante: muestra la base de datos, el controlador y, los elementos que contiene la vista; con los siguientes descriptores de acción: patron Registro, cardinalidad de 1 . . n, e información Ascendente.

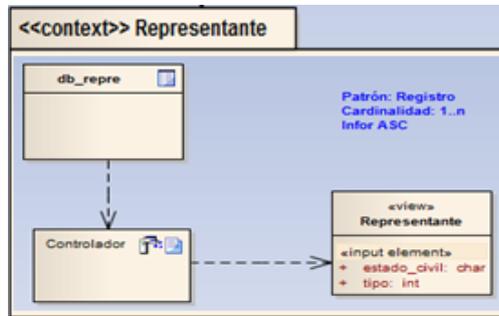


Figura IV. 36: Contexto de Representante

Fuente: Autor

Contexto Padres: muestra la base de datos, el controlador y, los elementos que contiene la vista; con los siguientes descriptores de acción: patron Registro, cardinalidad de 1..1, e información Ascendente.

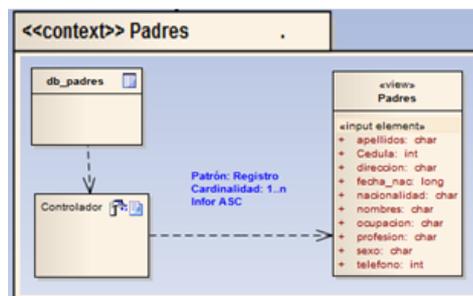


Figura IV. 37: Contexto de Padres

Fuente: Autor

4.1.2.2. Desarrollo de la Solucion

Capa de Presentación: Se desarrolla la interfaz principal mediante paginas HTML que se muestra en el navegador al ser compiladas, el cual va ser la presentación con la que el usuario interactúa, para facilitar la validación de los datos, la navegación y animar la interfaz del usuario se usa programas interpretes (JavaScript) y programas compilados (Applets de java).

Capa de Aplicación: llamado también lógica de Negocios que normalmente está basada en la codificación o programación de clases, funciones, controladores y vistas como se observa en la figura IV.38, para ello se reitera la utilización de la herramienta de desarrollo Netbeans 7.2.1 - Java Prime Face.

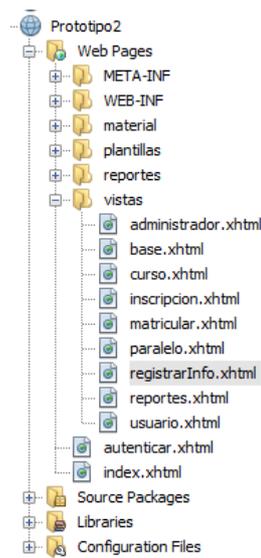


Figura IV. 38: Implementación del Prototipo OOWS

Fuente: Autor

Capa de Base de Datos: En esta capa se especifica el modelado de la base de datos con sus respectivas tablas y relaciones como se observó en la figura IV.23, que para este caso se toma el mismo diagrama presentado en la primera metodología por estar desarrollado en la misma herramienta DBMS PostgreSQL 9.2.

4.2. EJECUCIÓN DE LA EVALUACIÓN.

Luego de haber concluido los prototipos se procede a evaluar los indicadores presentados en el tercer capítulo, tomando los datos obtenidos durante el desarrollo del mismo.

Indicador 1: Evaluación en base al tipo de requisitos.

La siguiente evaluación se realiza en base a lo investigado en la ingeniería de requisitos, ya que el uso de una metodología permite conocer el tipo de requerimiento que necesita la aplicación.

Tabla IV. II: Evaluación del tipo de requisitos

TIPO DE REQUERIMIENTO	Prototipo UWE	Prototipo OOWS
R. Datos	SI	SI
R. Interfaz	NO	SI
R. Personalización	NO	SI
R. Navegacionales	SI	SI
R. Transaccionales	SI	SI
R. No Funcionales	SI	SI

Fuente: Autor

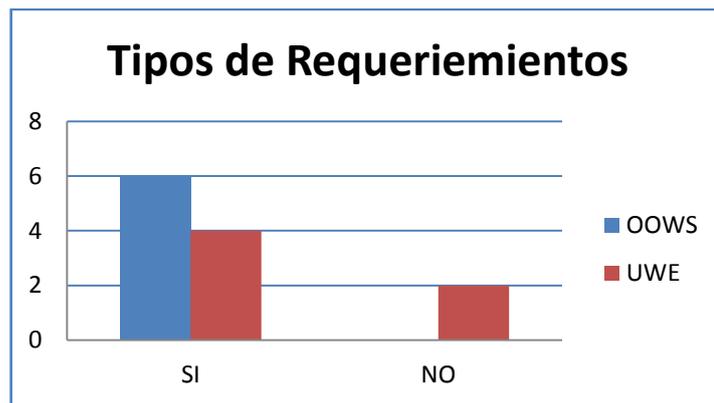


Figura IV. 39: Gráfica de resultados de Tipos de requerimientos

Fuente: Autor

En la figura IV.39 se puede observar que la metodología OOWS cumple con todos los tipos de requisitos, obteniendo como resultado el número máximo de SI.

Indicador 2: Evaluación en base a la propuesta funcional

Los indicadores en base a la propuesta funcional ayudan a determinar si existen herramientas propias para el diseño y el desarrollo, o por el contrario la metodología usa herramientas

generales de diseño de software (Tabla IV.III), y para ello se establece una tabla de valoración a escala de 1 a 3 (tabla IV. IV).

Tabla IV. III: Evaluación de la propuesta funcional

	UWE	OOWS
Notación del Dominio	UML	UML
Notación de Navegación	UML	UML + propia
Notación de Presentación	UML	XML(DPA)

Fuente: Autor

Escala de valoración 1-3: Se determina los valores de acuerdo a la importancia de la herramienta, es decir, si es cualquier herramienta común tiene el valor mínimo, pero si la herramienta es propia del método adquiere el valor máximo.

Tabla IV. IV: Escala de valoración de la propuesta funcional

Otro	1
Otro + PROPIA	2
Propia	3

Fuente: Autor

En base a lo detallado anteriormente se realiza la comparación según la propuesta funcional de este indicador como se observa en la Tabla IV.V.

Tabla IV. V: Comparación según propuesta funcional

METRICA	UWE	OOWS
Notación del Dominio	1	1
Notación de Navegación	1	2

Notación de Presentación	1	3
PROMEDIO	1	2

Fuente: Autor

$$UWE = 1 * 100 / 3 = 33.33\%$$

$$OOWS = 2 * 100 / 3 = 66.67\%$$

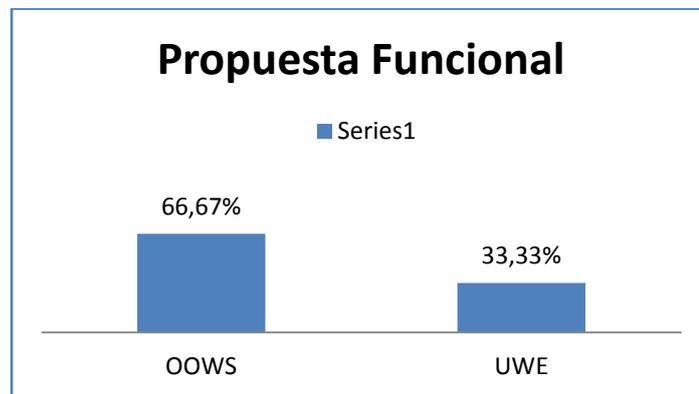


Figura IV. 40: Gráfica del resultado de la Propuesta Funcional

Fuente: Autor

Para la propuesta funcional se puede observar (figura IV.40), ambas metodologías usan notación UML, sin embargo a la hora de utilizar notación de navegación y presentación la metodología OOWS presenta herramientas propias para el diseño, por consecuencia cambia el puntaje de valoración y recibe un mayor puntaje.

Indicador 3: Evaluación del tiempo de diseño

Para determinar el tiempo utilizado en el diseño de cada metodología se toma el valor promedio calculado por cada diagrama, como se indica en el anexo 3, y se multiplica por el número de diagramas de cada metodología.

Tabla IV. VI: Evaluación del tiempo de diseño

Metodología	N° de Diagramas	Valor promedio	Total
UWE	15	0,82	12,13 horas
OOWS	12	0,82	9,81 horas

Fuente: Autor

En la figura IV.41 se puede observar que la metodología UWE tiene el valor de 12 horas y 13 minutos que corresponde al tiempo empleado en el diseño, mientras que para la metodología OOWS se requiere menos tiempo que es de 9 horas, es decir se tiene como resultado mayor productividad en la metodología con menor tiempo.

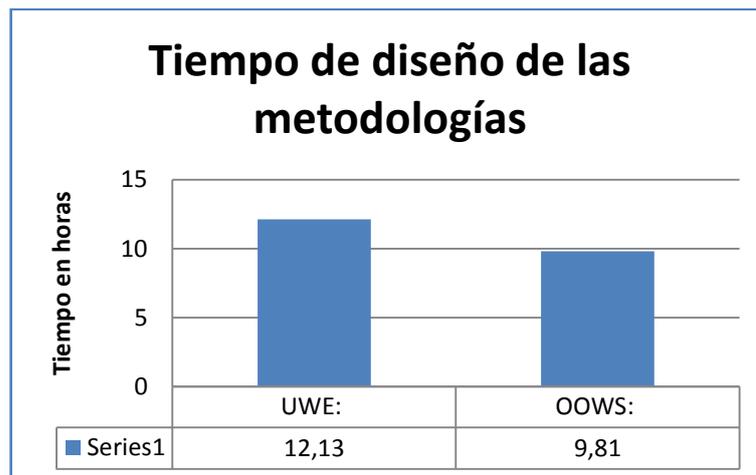


Figura IV. 41: Grafica del resultado del Tiempo de diseño

Fuente: Autor

Indicador 4: Evaluación del modelo de producto.

Para llevar a cabo una correcta evaluación es necesario determinar una tabla de valoración, tanto para el modelo de producto como para el modelo de proceso.

Escala de valoración por cumplimiento 4 – 1

Tabla IV. VII: Escala de valoración para el modelo de producto y proceso

Nominación	Valor
TOTALMENTE	4
PARCIALMENTE	3
ESCASAMENTE	2
NADA	1

Fuente: Autor

Modelo del producto: A continuación se realiza la comparación entre ambas metodologías de acuerdo a la escala de valoración.

Tabla IV. VIII: Evaluación del modelo de producto

METRICA	UWE	OOWS
El modelo del producto es implícito	3	1
El modelo del producto es explícito	1	4
La notación del producto es Formal (fórmulas matemáticas)	1	1
La notación del producto Semi-formal (Cuadros, gráficos)	3	4
La notación del producto Informal (Textual)	2	2
El modelo está Orientado a objetos	4	4
El modelo está Orientado a procesos	2	3
El modelo está basado en ejemplos	1	1
La orientación del modelo es Indefinida	1	1
La Perspectiva del producto es Funcional	3	4
La Perspectiva del producto Tecnológica	2	3
La Perspectiva del producto Estructural	3	3
TOTAL	26	31

Fuente: Autor

De acuerdo a la valoración por categoría nominal, se realiza la sumatoria por cada metodología y se halla un valor total para cada una, el cual se puede observar en la figura IV.42, que la metodología OOWS obtiene el mayor puntaje con relación a la metodología UWE.

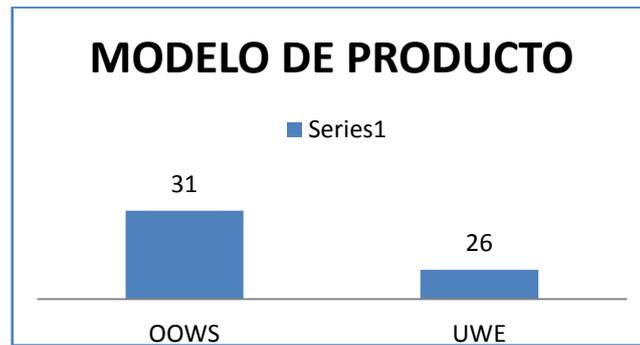


Figura IV. 42: Gráfica del resultado de Modelo de producto

Fuente: Autor

Indicador 5: Evaluación del modelo de proceso.

Tabla IV. IX: Evaluación del modelo de proceso

METRICAS	UWE	OOWS
El modelo del proceso está orientado al producto	3	3
El modelo del proceso está Orientado a la actividad	3	2
El modelo del proceso está Orientado a la decisión	1	1
El enfoque de Desarrollo es preventivo	2	3
El enfoque de desarrollo es evolutivo	2	3
La claridad del proceso tiene una estructura bien definida	3	4
No define una estructura	1	1
La estructura del proceso tiene fases o pasos	4	3
La estructura del proceso tiene tareas	2	4

Modo de ejecución del proceso es Interactivo	2	4
Modo de ejecución del proceso es Secuencial	2	2
TOTAL	25	30

Fuente: Autor

La sumatoria de los valores de acuerdo a la tabla nominal por cada metodología está representada en la figura IV.43, que da como resultado a la metodología OOWS con el mejor valor en la evaluación.

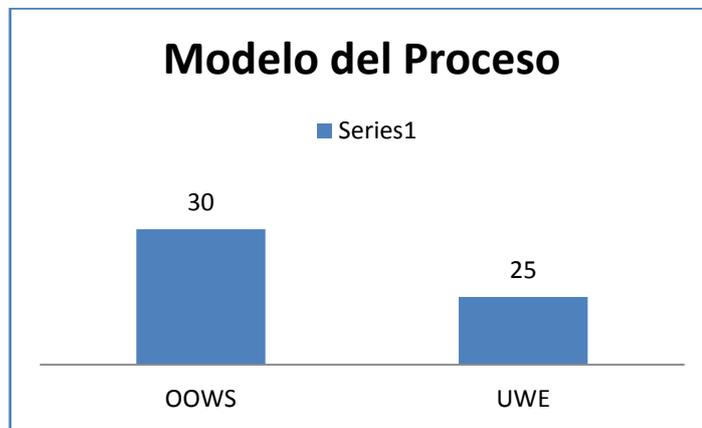


Figura IV. 43: Gráfica del resultado de Modelo de proceso

Fuente: Autor

Indicador 6: Evaluación por puntos de función.

Se propone una estimación previa a la comparación para tomar como referencia el número de líneas de código, el costo, la producción y el esfuerzo. El desarrollo de dicha estimación se realiza en el software COCOMO, con el ingreso de los puntos de función (figura IV.44) el cual está respaldado en el anexo 2.

SLOC Input Dialog - Matricula

Sizing Method:
 SLOC
 Function Points
 Adaptation

Breakage:
 % of code thrown away due to requirements volatility
 BRAK: 0.00

Module Size in Function Points
 Language: Object-oriented

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	2	0	2	18
Outputs	0	1	2	19
Files	1	2	1	42
Interfaces	0	1	3	37
Queries	0	2	2	20
Total Unadjusted Function Points				136
Equivalent Total in SLOC				4352

Buttons: OK, Cancel, Help

Figura IV. 44: Puntos de función en COCOMO

Fuente: Autor

El calculo de la estimación por puntos de función se puede determinar antes del diseño de la arquitectura denominado Early Design, en el cual muestra las líneas de código, el esfuerzo, costo y producto en tres aspectos como son : optimista, normal y pesimista como se puede observar en la figura IV.45 .

Project Name: Parvulario
 Scale Factor: [] Schedule: []
 Development Model: Early Design

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	NCM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Matricula	F: 4352	390.00	0.55	14.6	8.0	545.5	3111.45	0.7	1.1	0.0

Total Lines of Code:	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
4352	Optimistic	5.3	6.2	814.2	2084.67	0.5	0.9	
	Most Likely	8.0	7.1	545.5	3111.45	0.7	1.1	0.0
	Pessimistic	12.0	8.0	363.7	4667.18	1.1	1.5	

Project File: D:\TESIS_1\Parvulario1.est Is Loaded

Figura IV. 45: Estimación de productividad de diseño temprano en COCOMO

Fuente: Autor

También se determina la estimación de esfuerzo después del diseño de la arquitectura denominado Post Architecture como se observa en la figura IV.46, para dar a conocer la diferencia entre los valores determinados en los parámetros del software; esfuerzo, líneas de código, producto y costo.

The screenshot shows a software window titled 'D:\TESIS_1\Parvulario1.est - USC-COCOMO II1999.0'. The 'Project Name' is 'Parvulario' and the 'Development Model' is 'Post Architecture'. A table displays the following data for the 'Matricula' module:

Y	Module Name	Module Size	LABOR Rate (\$/month)	NAF	NCM Effort DEV	EST Effort DEV	PRGD	COST	INST COST	Staff	RISK
	Matricula	F:4352	390.00	0.57	14.6	8.3	523.3	3243.36	0.7	1.2	0.0

Below the table, a summary section shows 'Total Lines of Code' as 4352. It also provides estimated values for Effort, Sched, PRGD, COST, INST, Staff, and RISK under three scenarios: Optimistic, Most Likely, and Pessimistic.

	Estimated	Effort	Sched	PRGD	COST	INST	Staff	RISK
Optimistic	6.7	6.7	654.1	2594.69	0.6	1.0		
Most Likely	8.3	7.2	523.3	3243.36	0.7	1.2	0.0	
Pessimistic	10.4	7.7	418.6	4054.21	0.9	1.4		

Figura IV. 46: Estimación de productividad en Post Arquitectura

Fuente: Autor

Indicador 6.1: Puntos de Función en metodología UWE

En este indicador se realiza la evaluación de cada uno de los prototipos contabilizando el tiempo en horas y líneas de código para hallar la cantidad de software desarrollado, el tiempo de entrega y el costo.

- **Cantidad de software desarrollado por unidad de tiempo de trabajo.**

Número de puntos de función desarrollados: **237 LÍNEAS DE CÓDIGO**

Tiempo empleado: **1,4 HORAS**

Productividad TI = Total PF Completos / Total de esfuerzo en TI

Productividad TI= 237 / 1,4 = 169,27

- **Ritmo de entrega o Capacidad de desarrollo por unidad de tiempo transcurrido.**

Número de puntos de función desarrollados: **3921 LINEAS DE CODIGO**

Tiempo transcurrido: **47 HORAS**

Ritmo de entrega = #PuntoFunción Completos /Tiempo transcurrido

Ritmo de entrega=3921 / 47 = 83,42

- **Funcionalidad entregada por unidad de coste.**

Número de puntos de función desarrollados:**3921LINEAS DE CODIGO**

Coste total: **150 DOLARES**

Funcionalidad-coste = #PuntosFunción Completos desarrollados / Coste total

Funcionalidad-coste =3921 / 150 = 26,14

Indicador 6.2: Puntos de función de la metodología OOWS

Se realiza la evaluación del prototipo OOWS, que busca calcular el tiempo en horas y líneas de código para hallar la cantidad de software desarrollado, el tiempo de entrega y el costo.

- **Cantidad de software desarrollado por unidad de tiempo de trabajo.**

Número de puntos de función desarrollados: **247,75 LÍNEAS DE CÓDIGO**

Tiempo empleado: **1 HORA**

Productividad TI = Total PF Completos / Total de esfuerzo en TI

Productividad TI=247,75

- **Ritmo de entrega o Capacidad de desarrollo por unidad de tiempo transcurrido.**

Número de puntos de función desarrollados:**3681LINEAS DE CODIGO**

Tiempo transcurrido:**42 HORAS**

Ritmo de entrega = #PuntoFunción Completos /Tiempo transcurrido

Ritmo de entrega=3681 / 42 = 87,64

- **Funcionalidad entregada por unidad de coste.**

Número de puntos de función desarrollados:**3681 LINEAS DE CODIGO**

Coste total = **125 DOLARES**

Funcionalidad-coste = #PuntosFunción Completos desarrollados / Coste total

Funcionalidad-coste = **3681 / 125 = 29,45**

4.3. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

En esta etapa se lleva a cabo el análisis de cada uno de los indicadores, en el cual se interpreta los resultados finales que nos ayuda a determinar la metodología seleccionada.

Indicador 1: Evaluación en base al tipo de requisitos

La metodología OOWS cumple con la determinación de todos los requisitos, obteniendo SI en los 6 tipos de requisitos, el cual facilita al desarrollo de una aplicación funcional y con mejor productividad, mientras que UWE obtiene dos negativos y cuatro positivos.

Obteniendo los siguientes peso:

$$\text{UWE: } 4 * 100 / 6 = 66,67\%$$

$$\text{OOWS: } 6 * 100 / 6 = 100\%$$

Indicador 2: Evaluación en base a la propuesta funcional

Cuando se trata de propuesta funcional, se observa la homogeneidad de la notación utilizada por las metodologías, gran parte UML, donde solamente OOWS utiliza notación propia. Dentro de todos los conjuntos de metodologías, las propuestas funcionales son las que presentan una mayor cantidad, gracias a la comparación se puede notar que las aquí analizadas son muy parecidas, donde UML es el factor principal para esta igualdad en las características estudiadas. Esto puede deberse al efecto positivo que presenta UML como estándar para el modelado,

especialmente dirigido hacia la funcionalidad donde la correcta utilización de diagramas (como casos de uso) y el agregado de extensiones para incorporar elementos web puede entregar una herramienta potente para el modelado de este tipo de aplicaciones.

Finalmente los valores de la evaluación demuestran que la metodología OOWS tiene herramientas propias específicas para diseño y desarrollo web, el cual que incrementan la productividad a la hora del desarrollo, determinando de acuerdo a la evaluación los siguientes pesos:

UWE: cumple 1 parámetro de 3

$$1 * 100 / 3 = 33,33\%.$$

OOWS: cumple 2 parámetros de 3

$$2 * 100 / 100 = 66,67\%.$$

Indicador 3: Evaluación del tiempo de diseño

La metodología UWE obtiene el menor valor por poseer mayor número de diagramas en el diseño de su metodología, mientras que OOWS requiere menos tiempo para realizar su diseño. Es decir se toma como pesos finales la relación inversa que es a más tiempo menos productividad y a menos tiempo más productividad como se detalla a continuación:

Tiempo de OOWS: 9,81 horas

$$\text{Porcentaje: } 9,81 * 100 / 12,13$$

Peso: 80,87 %

Tiempo de UWE: 12,13 horas

$$\text{Porcentaje: } 12,13 * 100 / 9,81$$

Peso: 61,82 %

Indicador 4 y 5: Evaluación del modelo de producto y de proceso

El resultado para estos indicadores proviene del mismo estándar de evaluación, asigna valores a cada parámetro de acuerdo a la tabla establecida luego efectúa la sumatoria por cada tabla y por cada metodología como se explica a continuación:

Modelo del Producto

$$\text{OOWS: } \Sigma (x) = 31 \quad \rightarrow \quad 31 * 100 / 33 = 93,94\%$$

$$\text{UWE: } \Sigma (y) = 26 \quad \rightarrow \quad 26 * 100 / 33 = 78,79\%$$

Modelo de Proceso

$$\text{OOWS } \Sigma (x) = 30 \quad \rightarrow \quad 30 * 100 / 33 = 90,91\%$$

$$\text{UWE } \Sigma (y) = 25 \quad \rightarrow \quad 25 * 100 / 33 = 75,76\%$$

Indicador 6: Evaluación por Puntos de Función

Se realiza el análisis en base la estimación propuesta en COCOMO en relación con los puntos de función hallados en cada metodología.

Indicador 6.1 y 6.2: Puntos de Función en METODOLOGÍAS UWE y OOWS

Tabla IV. X: Evaluación en pesos de puntos de función

	UWE	Pesos	OOWS	Pesos
Productividad	169,27	46,54%	247,75	68,12%
Líneas de código	3921	90,10%	3681	84,58%
Ritmo de entrega	83,42 (código por hora)	83,42%	87,64 (código por hora)	87,64%

Funcionalidad- Coste	26,14 (código por 1 dólar)	47%	29,45 (código por 1 dólar)	52%
PROMEDIO		66,77%		73,10%

Fuente: Autor

En comparación con lo estimado en la herramienta COCOMO, total de líneas de código: 4352 líneas, con producción de 363,70 y un costo de \$ 4673,45.

Se obtiene como resultado que con la utilización de una metodología se logra una mayor productividad con un tiempo apropiado, deduciendo así que para la metodología OOWS se obtiene el mayor valor por tiempo de entrega.

4.4. COMPROBACIÓN DE LA HIPÓTESIS DE INVESTIGACIÓN

Para finalizar el estudio comparativo de las metodologías UWE y OOWS que permiten obtener la mayor productividad en el desarrollo de una aplicación web, se puede observar los valores comparativos por cada indicador.

Tabla IV. XI: Evaluación de pesos para la comprobación de hipótesis

INDICADORES	PESO UWE	PESO OOWS
<i>I1:</i> Evaluación en base al tipo de requisitos	66,67%	100%
<i>I2:</i> Evaluación en base a la propuesta funcional	33,33%	66,67%
<i>I3:</i> Evaluación del tiempo de diseño	61,82 %	80,87 %
<i>I4:</i> Evaluación del modelo de producto	78,79%	93,94%
<i>I5:</i> Evaluación del modelo de proceso	75,76%	90,91%

I6: Evaluación por Puntos de Función	66,77%	73,10%
TOTAL	383,14	505,29
PROMEDIO	63,86%	84,22%

Fuente: Autor

Luego del análisis respectivo de los pesos por cada indicador se observa los resultados de la metodología de mayor puntuación, demostrado en la figura IV.47.

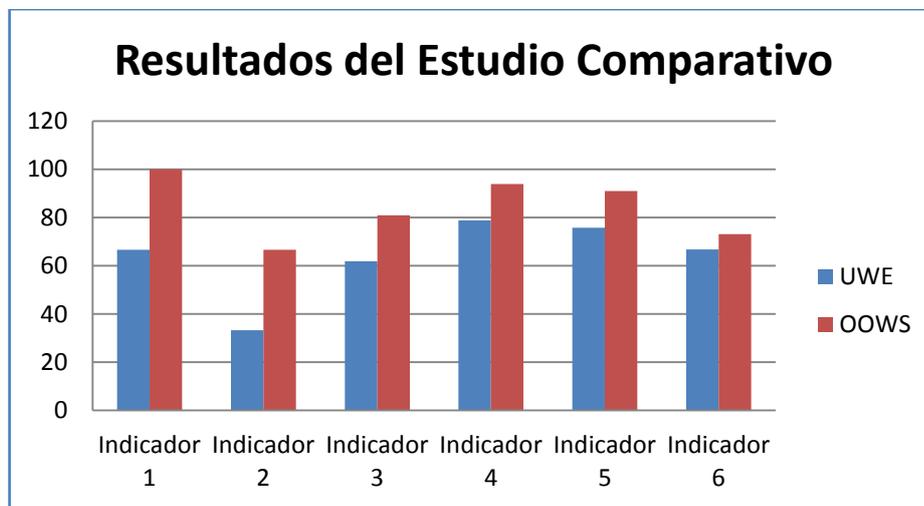


Figura IV. 47: Gráfica de resultados del estudio comparativo

Fuente: Autor

Finalmente para concluir la investigación se comprueba la hipótesis mediante el cálculo de la distribución Normal de T Student tomando como datos los pesos determinados en el análisis.

Distribución T Student

Dados los siguientes datos (tabla IV.XI) con los pesos finales de los indicadores planteados para la comparación de las metodologías UWE y OOWS se resuelve calcular el intervalo de confianza del 95% , el mismo que comprueba la hipótesis planteada.

Tabla IV. XII: Datos para el cálculo de T Student

PESO UWE	PESO OOWS
66,67	100
33,33	66,67
61,82	80,87
78,79	93,94
75,76	90,91
66,77	73,10

Fuente: Autor

Fórmulas de Media y Desviación Estándar o Típica

La fórmula de Media nos permite calcular el promedio de los resultados finales y, la desviación típica llamada también la varianza que se produce entre los valores obtenidos, dichos valores permiten el cálculo del método de Distribución T Student.

Media aritmética	$\bar{x} = \frac{\sum x_i n_i}{N}$
Varianza (s^2) y desviación típica (s)	$s^2 = \frac{\sum x_i^2 n_i}{N} - \bar{x}^2$; $s = \sqrt{\frac{\sum x_i^2 n_i}{N} - \bar{x}^2}$

Figura IV. 48: Formulas de Media y Desviación Estándar

Fuente: Autor

Se presenta la formula utilizada para el cálculo del intervalo de confianza (figura IV.49), entre los resultados obtenidos, la misma que ayuda a la comprobación de la hipótesis.

$$\mu \in \left(\bar{x} - t_{\left(\frac{\alpha}{2}, n-1\right)} \frac{S}{\sqrt{n}}, \bar{x} + t_{\left(\frac{\alpha}{2}, n-1\right)} \frac{S}{\sqrt{n}} \right)$$

Figura IV. 49: Formula de Intervalo de Confianza

Fuente: Autor

METODOLOGÍA UWE

Media = 63,86

$$s = \sqrt{\frac{(66,67 - 63,86)^2 + (33,33 - 63,86)^2 + (61,82 - 63,86)^2 + (78,79 - 63,86)^2 + (75,76 - 63,86)^2 + (66,77 - 63,86)^2}{6}}$$

$$s = \sqrt{\frac{7.9 + 932.08 + 4.16 + 222.90 + 141.61 + 8.47}{6}}$$

s = 36,29

En la tabla se tiene que t (0.025) con 4 grados de libertad es de 2.7765, donde el intervalo de confianza de 95% para μ es:

$$63,86 - (2.776) * (36,29 / \sqrt{6}) < \mu < 63,86 + (2.776) * (36,29 / \sqrt{6})$$

Con un nivel de confianza del 95% se sabe que el promedio de productividad para la metodología UWE están entre:

$$\mathbf{63.86 - 93.94 < \mu < 63.86 + 93.94}$$

$$\mathbf{- 30.08 < \mu < 157.8}$$

Como se puede observar el rango para el nivel de confianza de UWE está dentro de un valor negativo, lo que significa que dicha metodología está en desaprobación.

Metodología OOWS

Media = 84,22

$$s = \sqrt{\frac{(100 - 84,22)^2 + (66,67 - 84,22)^2 + (80,87 - 84,22)^2 + (93,94 - 84,22)^2 + (90,91 - 84,22)^2 + (73,10 - 84,22)^2}{6}}$$

$$s = \sqrt{\frac{249 + 308 + 11,22 + 94,48 + 44,76 + 123,65}{6}}$$

$$s = 28,83$$

En la tabla se tiene que $t(0.025)$ con 4 grados de libertad es de 2.7765, donde el intervalo de confianza de 95% para μ es:

$$84,22 - (2.776) * (28,83 / \sqrt{6}) < \mu < 84,22 + (2.776) * (28,83 / \sqrt{6})$$

Con un nivel de confianza del 95% se sabe que el promedio de productividad para la metodología OOWS están entre:

$$84,22 - 32,67 < \mu < 84,22 + 32,67$$

$$51,55 < \mu < 116,89$$

De acuerdo al resultado obtenido por medio de la Distribución Normal T Student se llega a la conclusión que el intervalo de confianza para la metodología OOWS mantiene un rango positivo mayor 50, considerado a muy bueno por tanto como resultado final se tiene que: tanto para el cálculo objetivo por pesos como el cálculo estadístico se demuestra que la metodología OOWS ayuda a mejorar la productividad en el desarrollo de una aplicación web, comprobando así la hipótesis establecida.

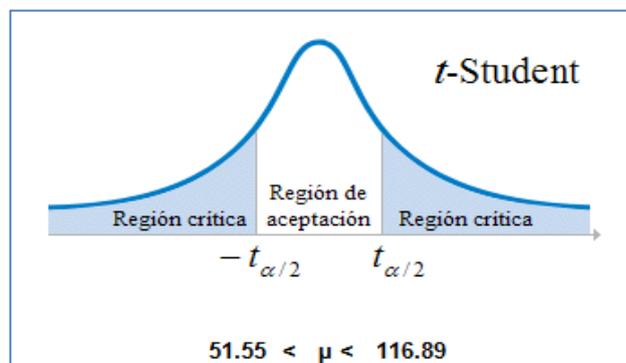


Figura IV. 50. Intervalo de confianza que aprueba la hipótesis

Fuente: Autor

CAPÍTULO V

APLICACIÓN WEB PARA EL PARVULARIO POLITÉCNICO

Se desarrolla la aplicación web dirigida al Parvulario Politécnico, mediante la aplicación de la metodología seleccionada OOWS.

5.1. GESTIÓN DEL PROYECTO

En este capítulo se realiza el desarrollo del Sistema web para el Centro Parvulario Politécnico, el cual se describe brevemente como está constituido y cuáles son sus funciones.

5.1.1. ESTRUCTURA ORGANIZACIONAL

El centro Parvulario Politécnico es una entidad que forma parte de la ESPOCH como un servicio más que brinda a la comunidad tanto dentro como fuera de la institución, figura V.51, sin embargo mantiene una administración independiente ya que consta de autoridades, personal administrativo y de Servicio estructurado en el organigrama institucional como se puede observar en la figura V.52.

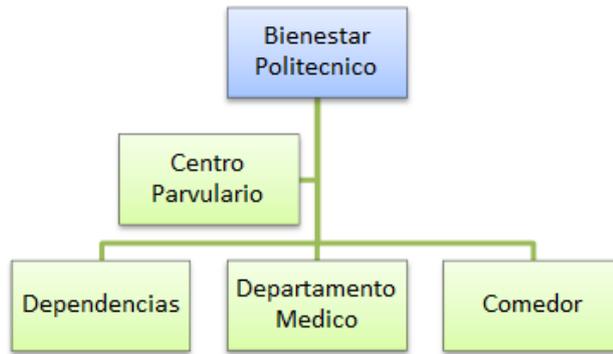


Figura V. 51: Estructura Organizacional

Fuente: Autor

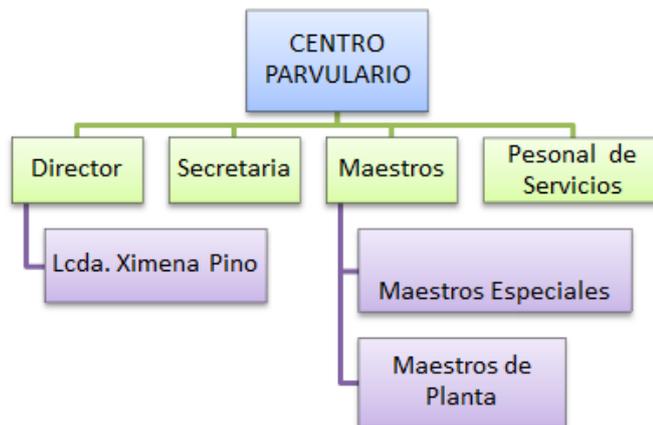


Figura V. 52: Organigrama Institucional

Fuente: Autor

5.1.2. FUNCIONALIDAD

Como principal autoridad se tiene a la Sra. Directora Lcda. Ximena Pino que supervisa y autoriza las actividades que se lleva a cabo dentro del Parvulario, sin embargo la administración académica la lleva la Sra. Secretaria Lcda. Tania, persona que se tomaría como administrador del sistema parvulario, quien se encarga de inscripciones, matriculas, registro de cobros, emisión de reportes y más. La otra funcionalidad es la del maestro que tiene a cargo una lista curso a quienes pertenezcan a dichos paralelos debe realizar la evaluación correspondiente.

5.1.3. DESCRIPCIÓN DEL SISTEMA

El sistema como tal debe llevar un control de inscripciones, matrículas y evaluación de alumnos, para ello se ha desarrollado dos perfiles de usuarios el administrador y el maestro.

USUARIO 1: Administrador parvulario realiza el registro de año lectivo, nuevas inscripciones, seguidamente el registro del alumno, padres de familia y el tipo de representante, luego procede a la matriculación del alumno.

También realiza el registro de datos de los profesores, materias, cursos, asignación de paralelos y registro de pagos, de tal manera que puede emitir informes de los datos almacenados en forma de reportes.

El administrador como tal posee la autorización de cambiar de rol, es decir ingresa como administrador y pasa a ser maestro, lo que significa que solo el administrador tiene el control completo del sistema, en donde tiene facilidad de obtener un reporte en el momento que lo requiera.

USUARIO 2: Evaluador parvulario o maestro, ingresa con su respectivo usuario y contraseña para tener acceso a la pantalla principal que consta del menú Evaluación y reportes de la evaluación cualitativa.

5.1.3.1. Perspectiva del producto

El sistema se desarrolló en el lenguaje de programación Java con la ayuda del framework JSF, el mismo que se publicó en el servidor de la ESPOCH, permitiendo su acceso desde cualquier navegador web (Mozilla Firefox, Google Chrome, Internet Explorer, Opera, etc.), de la misma manera para el almacenamiento de la información se utilizó el gestor de base de datos PostgreSQL, por su rapidez y bajo consumo de recursos, además tanto Java como PostgreSQL no incluyen costos adicionales de licencia por ser software libre.

En la figura V.53 se muestra como el administrador una vez que tiene acceso al sistema puede administrar todos los módulos incluyendo el cambio de rol para realizar las evaluaciones, la obtención de reportes.

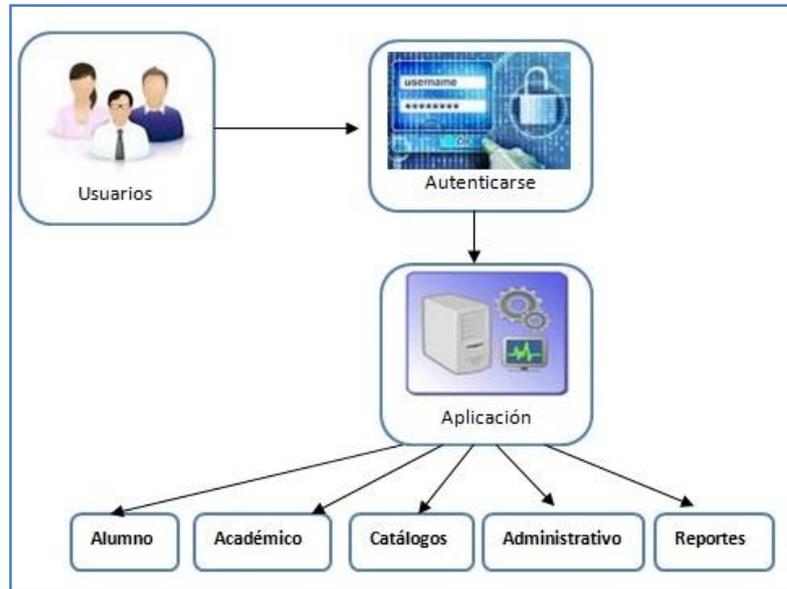


Figura V. 53: Perspectiva del producto

Fuente: Autor

5.1.3.2. Diseño de interfaces

Todas las interfaces utilizadas en el sistema deben ser amigables, sencillas, completamente gráficas, intuitivas y contar con soporte para la operación y navegación a través de un mouse, para ello se diseñan las pantallas previas de acceso al sistema (figura V.54), pantalla principal (figura V.55), módulo matrícula y módulo de reportes.

Pantalla de Acceso: Esta interfaz está basada en 2 marcos que son uno para el inicio de sesión y el otro para mostrar información.



Figura V. 54: Diseño de la pantalla de acceso o inicio de sesión

Fuente: Autor

Página Principal: La interfaz principal debe contar con tres marcos o frames:

- El primero debe contener un menú principal que contenga accesos directos hacia todas las secciones y subsecciones, debe existir una clara separación de los vínculos dependiendo de su función.
- El segundo marco debe contener el nombre y el logo de la empresa, el nombre del sistema y los datos del usuario activo.
- El tercer marco contiene el área de trabajo, es decir es un contenedor donde se presentaran todas las interfaces de ingreso y visualización de datos.



Figura V. 55: Diseño de la pantalla principal

Fuente: Autor

Pantalla de Matrícula: La interfaz de matrícula (figura V.56) está diseñada con estándar del número de frames y su ubicación para las demás pantallas el cual contiene un frame principal para el para mostrar los datos que se vayan almacenando o tengan selección, el cual lleva dos botones, uno para el ingreso y otro para la actualización de datos.

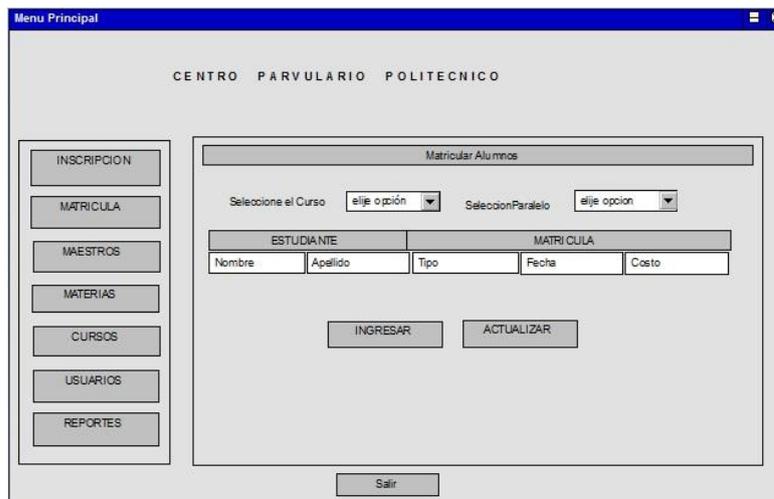


Figura V. 56: Diseño de pantalla de matrícula

Fuente: Autor

Seguidamente una vez seleccionado alguno de los botones se muestra el frame que contiene los campos respectivos para cada entidad.

5.2. DESARROLLO DE LA METODOLOGÍA OOWS

De acuerdo a la investigación realizada en el capítulo anterior se presenta la propuesta del desarrollo de una aplicación web para el sistema parvulario aplicando la metodología OOWS que proporciona mejor productividad, es decir se realiza paso a paso el desarrollo del diseño de la arquitectura y la programación, cumpliendo con sus procesos respectivos.

5.2.1. ESPECIFICACIÓN DEL SISTEMA

Esta etapa es la parte inicial del diseño del sistema, la misma que cuenta con subprocesos que permiten determinar todos los tipos de requerimientos en una aplicación web.

5.2.1.1. Especificación de Requisitos

La especificación de requisitos usa notación UML, que recoge la funcionalidad en navegación, interfaz e interacción con los diferentes tipos de usuarios. Para ello se realiza la captura de requisitos y definición por tipo.

Captura de Requisitos: Se desarrolla la captura y definición de los requisitos mediante dos pasos importantes que son:

Técnica de Recolección de información: Se cuenta con la ayuda de la Licenciada Tania Maldonado, quien nos proporciona la información necesaria tanto en documentación como verbalmente, es decir se realiza la descripción del funcionamiento del parvulario, la realización de inscripciones, matriculas, registro de alumnos, padres de familia y representante, el ingreso de materias, datos de maestros tanto personales como profesionales, cursos y paralelos, forma de evaluaciones a los alumnos, registro de pagos tanto para personas particulares como politécnicos, permitiendo conocer todos aquellos datos con el respaldo de una copia de cada documento o formulario utilizado manualmente.

Diseño de diagramas de casos de uso: La información recolectada anteriormente se conceptualiza y se plasma en el diseño de diagramas de caso de uso como se representa en la figura V.57.

Diagramas de Casos de Uso: Describe como el usuario realiza las acciones en la aplicación, permitiendo definir los perfiles de acceso; el rol de Administrador Master tiene la autorización de administrar las cuentas de usuario y, mediante el cambio de rol realizar las funciones de administrador y profesor parvulario.

El rol de administrador parvulario tiene los permisos necesarios para realizar las transacciones del sistema, como el uso del módulo académico, catálogos, administrativo y reportes, así también puede utilizar el cambio de rol a profesor, que permite realizar las evaluaciones a los alumnos. Se determina también el último perfil que es el de profesor y, permite la evaluación de los alumnos.

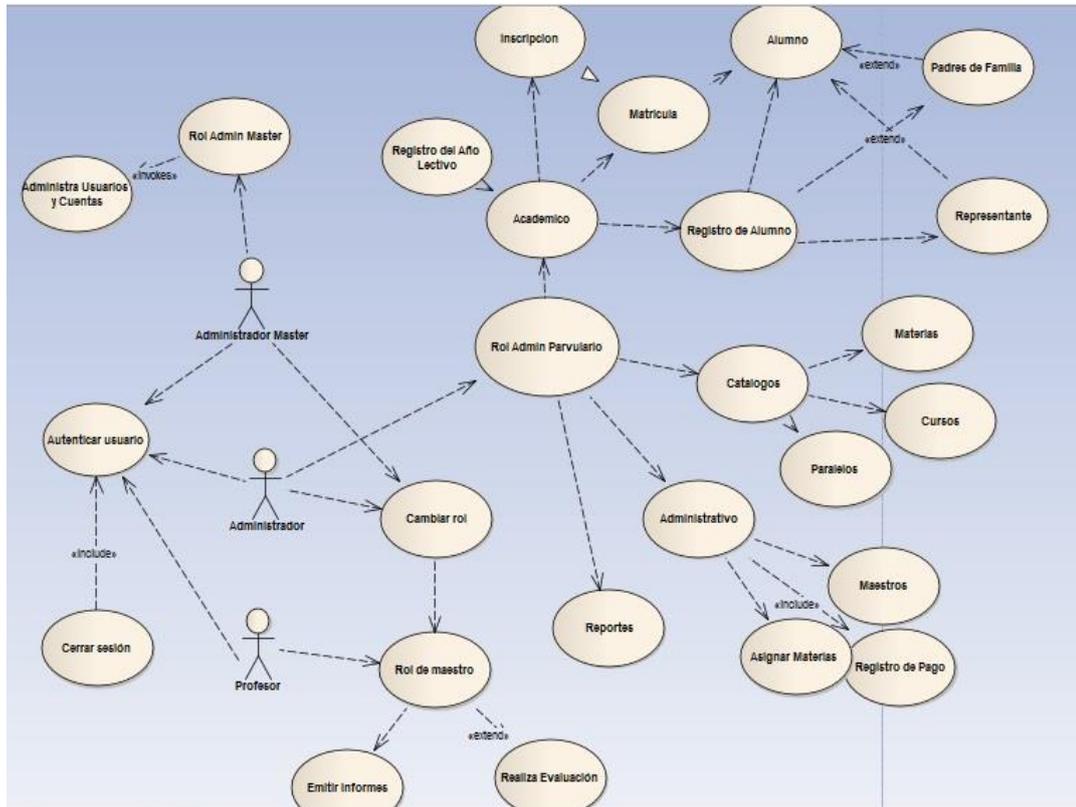


Figura V. 57: Diagrama de Caso de uso para la captura de requisitos

Fuente: Autor

Descripción de casos de uso: A continuación se muestra el detalle de información capturada en los diagramas de caso de uso para la especificación de los requisitos.

Tabla V. XIII: Descripción caso de uso para autenticación

DEFINICION	El sistema permite la autenticación del usuario
-------------------	---

ACTORES	Administrador (Directora y Secretaria)
PROPÓSITO	Mantener la seguridad del sistema mediante la identificación del usuario.
VISIÓN GENERAL	Cada usuario debe ingresar con su cuenta y clave.
TIPO	Primario

Fuente: Autor

Tabla V. XIV: Descripción caso de uso para el cambio de rol del administrador

DEFINICION	El sistema permite que el administrador pueda cambiar de rol
ACTORES	Administrador (Directora y Secretaria)
PROPÓSITO	El usuario administrador tiene la autorización de realizar las actividades de ambos perfiles
VISIÓN GENERAL	El administrador cambia su rol de acuerdo a la necesidad.
TIPO	Primario

Fuente: Autor

Tabla V. XV: Descripción del caso de uso para el registro de año lectivo

DEFINICION	Como administrador realiza el registro de un nuevo año lectivo
ACTORES	Secretaria
PROPÓSITO	Controlar el año lectivo presente de forma

	activa.
VISIÓN GENERAL	El ingreso de nuevo año lectivo debe ser activado y los periodos anteriores desactivados para realizar cualquier transacción o registro dentro del periodo presente.
TIPO	Primario

Fuente: Autor

Tabla V. XVI: Descripción de caso de uso de inscripción

DEFINICION	Como administrador realiza la inscripción de alumnos.
ACTORES	Secretaria
PROPÓSITO	Realizar la inscripción del alumno que postula para el ingreso.
VISIÓN GENERAL	Registro de datos básicos del alumno y tipo de representante
TIPO	Primario

Fuente: Autor

Tabla V. XVII: Descripción de caso de uso del registro de alumno

DEFINICION	Como administrador realiza el ingreso de los datos del alumno
ACTORES	Secretaria
PROPÓSITO	Llevar un control de los datos principales

	del alumno
VISIÓN GENERAL	Registrar los datos completos del alumno para confirmar su estancia en la institución.
TIPO	Primario

Fuente: Autor

Tabla V. XVIII: Descripción de caso de uso de matrícula

DEFINICION	Como administrador realiza el proceso de matriculación
ACTORES	Secretaria
PROPÓSITO	Llevar un control de los alumnos que forman parte de la institución.
VISIÓN GENERAL	Registro del curso y paralelo asignado.
TIPO	Primario

Fuente: Autor

Tabla V. XIX: Descripción de caso de uso del registro de maestro

DEFINICION	Ingresar datos personales y profesionales del maestro
ACTORES	Secretaria
PROPÓSITO	Llevar un control de maestros de la institución.
VISIÓN GENERAL	Registrar a cada maestro con sus datos tanto personales como profesionales.

TIPO	Primario
-------------	----------

Fuente: Autor

Tabla V. XX: Descripción de caso de uso de padres de familia

DEFINICION	Registro de datos de padres de familia
ACTORES	Secretaria
PROPÓSITO	Llevar un control de padres de familia de los alumnos
VISIÓN GENERAL	Ingresar los datos personales y controlar el número de representados a cargo.
TIPO	Primario

Fuente: Autor

Tabla V. XXI: Descripción de caso de uso del representante

DEFINICION	Registro de datos del representante
ACTORES	Secretaria
PROPÓSITO	Llevar un control de los datos del representante
VISIÓN GENERAL	Ingresar los datos del representante en caso de ser otro o realizar la búsqueda del padre o madre para asignarlo.
TIPO	Primario

Fuente: Autor

Tabla V. XXII: Descripción de caso de uso de cursos y paralelos

DEFINICION	Registro de cursos y paralelos
-------------------	--------------------------------

ACTORES	Secretaria
PROPÓSITO	Ingresar los cursos y paralelos que existe en la institución de acuerdo al nivel.
VISIÓN GENERAL	Ingreso del nombre de curso y paralelo.
TIPO	Primario

Fuente: Autor

Tabla V. XXIII: Descripción de caso de uso de materia

DEFINICION	Registro de materias
ACTORES	Secretaria
PROPÓSITO	Ingresar los nombres de las materias o asignaturas que imparten en la institución.
VISIÓN GENERAL	Ingreso del nombre de materia
TIPO	Primario

Fuente: Autor

Tabla V. XXIV: Descripción de caso de uso de registro de pago

DEFINICION	Como administrador realizar el registro de pagos (inscripción, matrícula y pensión)
ACTORES	Secretaria
PROPÓSITO	Controlar que cada alumno cancele los aranceles de la institución.
VISIÓN GENERAL	El pago realizado en la tesorería de la ESPOCH debe ser registrado con número de factura, para matriculas y pensiones

	mensuales en caso de ser particular.
TIPO	Primario

Fuente: Autor

Tabla V. XXV: Descripción de caso de uso de evaluaciones a alumnos

DEFINICION	Como maestro registrar las evaluaciones de los alumnos
ACTORES	Maestro
PROPÓSITO	Llevar un control general de las evaluaciones descriptivas de cada alumno.
VISIÓN GENERAL	El maestro debe registrar la evaluación descriptiva del alumno y la asignación de una calificación por quimestre y un informe al final.
TIPO	Primario

Fuente: Autor

Tabla V. XXVI: Descripción de caso de uso de crear nuevas cuentas

DEFINICION	Como administrador puede crear cuentas de usuario
ACTORES	Administrador (Directora - Secretaria)
PROPÓSITO	Agregar a los maestros como usuarios nuevos dentro del sistema.
VISIÓN GENERAL	El administrador tiene la autorización de registrar nuevos usuarios creando el usuario y contraseña para cada uno, y así

	logren tener acceso al sistema.
TIPO	Primario

Fuente: Autor

Tabla V. XXVII: Descripción del caso de uso de reporte de lista de alumno

DEFINICION	Como maestro revisar el listado de alumnos que tiene a cargo
ACTORES	Maestro
PROPÓSITO	Obtener el listado de alumnos por curso.
VISIÓN GENERAL	Seleccionar el curso y paralelo y generar el reporte de alumnos.
TIPO	Primario

Fuente: Autor

Tabla V. XXVIII: Descripción de caso de uso del reporte de registro de alumno

DEFINICION	Obtener un reporte del registro de alumno
ACTORES	Secretaria
PROPÓSITO	Acceder a los datos tanto del alumno como de sus padres en forma visual de una ficha informativa del registro del alumno.
VISIÓN GENERAL	Llevar un control de cada alumno con sus respectivos datos informativos, para administración interna.
TIPO	Primario

Fuente: Autor

Tabla V. XXIX: Descripción de caso de uso del reporte de registro de pagos por alumno

DEFINICION	Obtener un reporte del registro de pagos por alumno
ACTORES	Secretaria
PROPÓSITO	Revisar que los alumnos tengan los pagos al día.
VISIÓN GENERAL	Seleccionar el alumno y generar un reporte de pagos realizados por fecha.
TIPO	Primario

Fuente: Autor

Tabla V. XXX: Descripción de caso de uso del reporte orden de pago

DEFINICION	Obtener una orden de pago
ACTORES	Secretaria
PROPÓSITO	Emitir la orden de pago a los padres de familia, con la descripción correcta, nombre, valor, curso y número.
VISIÓN GENERAL	Generar un reporte de cada curso con sus respectivas materias.
TIPO	Primario

Fuente: Autor

Determinación de requisitos por tipo

De acuerdo al estudio de casos de uso realizado anteriormente se puede determinar todos los requerimientos que el sistema debe cumplir.

Requisitos Funcionales

- Autenticación del usuario (Administrador y Maestro)
- El administrador puede crear cuentas de usuario
- Como administrador realiza la inscripción de alumnos
- Como administrador realiza el proceso de matriculación
- Como maestro registrar las evaluaciones de los alumnos
- Como administrador realizar el registro de pagos (matrícula y pensión)
- Como maestro revisar el listado de alumnos que tiene a cargo
- Obtener un reporte del registro de cada alumno con sus respectivos datos y de sus padres.
- Obtener un reporte del registro de pagos por alumno
- Como maestro obtener el reporte de las evaluaciones del alumno

Requisitos de Datos

Los requisitos de datos, también denominados requisitos de contenidos, requisitos conceptuales o requisitos de almacenamiento de información, responden a la pregunta de qué información debe almacenar y administrar la aplicación.

- Registro del año escolar
- Registro de datos de los alumnos y padres de familia
- Ingresar datos de materias, cursos y paralelos
- Ingresar datos del representante
- Ingresar datos personales y profesionales del maestro
- Realizar actualizaciones de datos de alumnos y padres de familia
- Realizar actualizaciones de datos de materias, cursos y paralelos
- Realizar actualizaciones de datos de maestros

Requisitos de Interfaz

Los requisitos de interfaz, también llamados requisitos de interacción de usuario, responden a la pregunta de cómo va a interactuar el usuario con la aplicación.

- La aplicación debe gestionar de forma diferenciada los siguientes perfiles de usuarios: Administradores y Profesores.
- Permite la comunicación entre usuario y aplicación mediante mensajes de confirmación, advertencia o error en las transacciones.
- Los campos obligatorios deben estar marcados por un asterisco (*)
- Las pantallas deben tener el mismo formato (color, fuente, campos de ingreso, etc.).
- Todos los usuarios tienen un inicio de sesión, que deben autenticarse mediante usuario y contraseña.
- La aplicación requiere un cierre de sesión en donde la sesión se cerrará bien por la acción del usuario que la abrió o por un mecanismo automático basado en un determinado tiempo de espera en inactividad.

Requisitos de Personalización

- Como administrador permite cambiar de rol a maestro
- Permite visualizar el nombre del usuario en uso en la parte central de la interfaz.
- El usuario podrá escoger la plantilla base de presentación.
- La información se almacena por orden de ingreso y se administra en páginas.

Requisitos Navegacionales

- El sistema permite desplazarse de un módulo a otro sin mayor problema de perder el menú.
- Permite visualizar un menú para su navegación, de acuerdo al uso con una pestaña que minimiza y maximiza.

- Si se presenta un error el sistema da la alternativa de cancelar o salir de dicho modulo.

Requisitos No Funcionales

Amigabilidad

- El sistema proporcionara una interfaz gráfica amigable, fácil de utilizar e intuitiva para el usuario.

Disponibilidad

- Es sistema estará disponible las 24 horas ya sea para crear una solicitud, para cambiar el estado de las mismas, o para la generación de reportes con la finalidad de agilizar el proceso.

Fiabilidad

- El sistema es confiable ya que es sometido a continuas pruebas y validaciones para medir su grado de eficiencia.

Mantenibilidad

- Mantenimiento de la base de datos.

Seguridad

- El sistema tendrá un formulario de autenticación.
- Uso de sesiones de usuario.

5.2.1.2. Modelado Conceptual

Se realiza la captura de la estructura y el comportamiento del sistema desde los 3 puntos de vista fundamentales.

Vista Estructural: Diseño de la estructura conceptual de modulo matricula, aquí se describe las clases que intervienen con sus respectivos atributos y operaciones (figura V.58).

El diagrama conceptual consta de las siguientes clases Alumno que se relaciona con las clases, matricula, persona, padre:alumno, así mismo se tiene la clase Persona que se relación con las clases: Padres y maestros, la clase Maestro se relaciona con Paralelo, mientras que paralelo tiene relación con curso y materia asignada y, aquella se relaciona con Evaluación, también se tiene la clase Matricula relacionada con Año_lectivo y, aquella con Inscripción; las clases independientes son Representante poli, Costos y Registro pago.

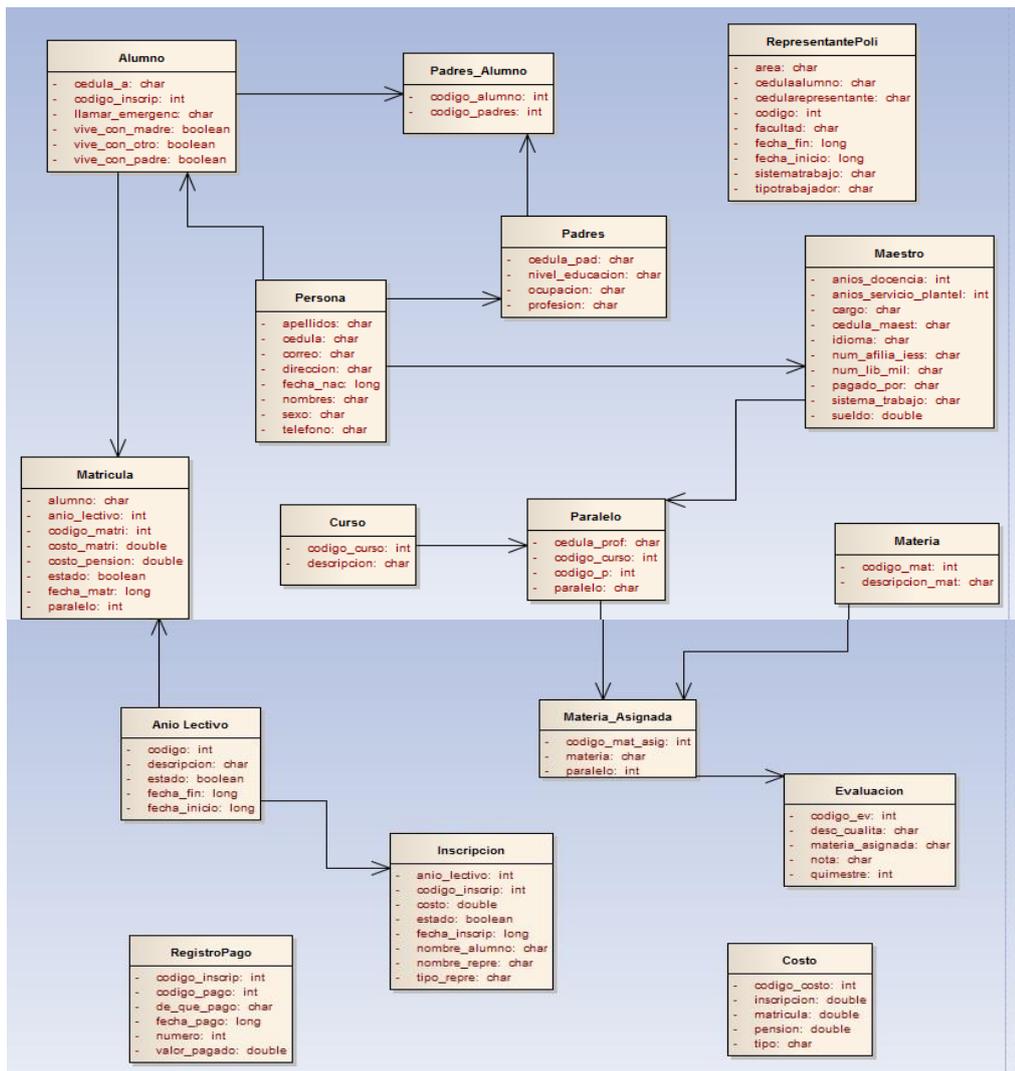


Figura V. 58: Vista estructural del modelo conceptual

Fuente: Autor

Vista Funcional: El esquema funcional está definido de la siguiente manera; el administrador inicia el proceso al ejecutar el sistema, se autentica con su nombre de usuario y contraseña e inmediatamente tiene acceso al control de los módulos expuestos en el menú, en el cual puede ingresar o actualizar datos. El menú consta del registro académico como es el año lectivo, inscripción, matrícula y registro de pagos, el registro del alumno que incluye datos del alumno, padres de familia, representante, los catálogos como son materias, cursos y paralelos, también consta de la administración de usuarios, maestros y asignación de materias.

Para cada registro de las entidades antes mencionadas se lleva a cabo un ingreso y actualización de datos, cabe recalcar que para ello existe un proceso independiente por tanto posee diferentes estados de acuerdo a la acción, un iniciar, un nuevo ingreso, una búsqueda, una espera, mensajes de error, un guardar, un actualizar, un cancelar y un salir o finalizar.

Vista Dinámica: La siguiente vista consta de diseño de diagramas de Transición de Estados y diagramas de Secuencia, para ello se realiza la descripción de acciones de los respectivos módulos.

PERFIL ADMINISTRADOR PARVULARIO

Módulo Gestión Académica

Objetivo: Controlar las operaciones transaccionales principales para el funcionamiento del sistema académico.

academ_año_lectivo: esta entidad almacena el periodo académico, descripción, estado, fecha que inicia y fecha que termina

academ_inscripción: realiza la operación de inscribir a los alumnos que desean guardar un cupo para ser matriculados en el parvulario.

academ_alumno: realiza el registro de datos del alumno y de sus padres, el usuario realiza el ingreso de los datos personales del alumno para llevar un control específico de cada uno; así también dentro de dichos datos consta el registro de padres de familia tanto del padre como de la madre y finalmente la asignación de un representante, la selección va a ser entre el padre, la madre que pueden ser asignados o en caso de ser otra persona un nuevo ingreso de datos.

academ_matrícula: Ingresa los datos completos de los alumnos que van a ingresar al parvulario, siempre y cuando exista una inscripción previa.

Módulo Administración

Objetivo: El siguiente modulo está considerado de alta importancia en el manejo, ya que el administrador puede llevar un control de usuarios, asignación de materias, evaluaciones, y maestros.

admin_registro_maestros: realiza el registro de los datos personales y académicos del maestro, así también debe llevar un control del tiempo y sistema de trabajo, el sueldo, las capacitaciones y más.

admin_registro_de_pagos: Este registro es indispensable para llevar un control de todos los pagos realizados ya sea matrícula, o pensión.

Módulo Catálogos

Objetivo: El administrador tiene la facilidad de controlar el ingreso y actualización de las materias, cursos y paralelos que presenta el centro parvulario.

cata_materia: se almacena el nombre de la materia que se dicta en cierto nivel.

cata_cursos: el usuario ingresa el nombre del curso para luego asignar las materias correspondientes de acuerdo al paralelo existente.

cata_paralelos: se almacena los paralelos existentes con la sintaxis de letras mayúsculas.

cata_asignar_materias: la asignación de materias está basada en el curso que se encuentre a cargo el maestro, ya que debe seleccionar el profesor, el curso y la materia que le corresponde dictar.

Modulo Reportes

Objetivo: Verificar los datos almacenados y transacciones realizadas en el sistema mediante un documento presentado como reporte ya sea en digital o en papel.

Repor_registro_de_alumnos_matriculados: la operación para la generación de un reporte es similar para todas las entidades, el usuario ingresa al módulo reportes y selecciona la acción Reporte Ingreso de alumno, seguidamente se lista todos los alumnos y, con la selección de uno de ellos en el icono pdf o reporte e inmediatamente se abre el archivo de los datos completos.

Repor_registro_de_pagos: El usuario debe ingresar al módulo de registrar pagos, luego seleccionar el alumno y generar el reporte de pagos seleccionando el icono o nombre del reporte.

PERFIL MAESTRO

Módulo Evaluación

maest_evaluación: la operación de evaluación a los alumnos es tarea exclusiva del maestro, sin embargo el administrador tiene autorización a ese modulo por la facilidad de cambio rol, de administrador a maestro y viceversa. Dicha evaluación se realiza por quimestres y de forma cualitativa, de acuerdo a una escala de valoración y de forma descriptiva para la emisión de reportes.

Repor_evaluaciones: este reporte es específico para los maestros que han realizado las respectivas evaluaciones, se puede generar por evaluación por materia, por curso y por alumno, se emite una evaluación parcial y una final.

PERFIL ADMINISTRADOR MASTER

Admin_crear_cuentas: el administrador tiene la potestad de crear cuentas de acceso al sistema tanto para el perfil de maestro como de administrador, de manera que los usuarios ya registrados obtengan un Nick o cuenta y una contraseña.

Diagrama de Transición de Estados

En la figura V.59 se muestra el proceso de cambios de un estado a otro mientras ejecuta el sistema, para realizar un ingreso nuevo, para actualizar, realizar búsquedas entre otros.

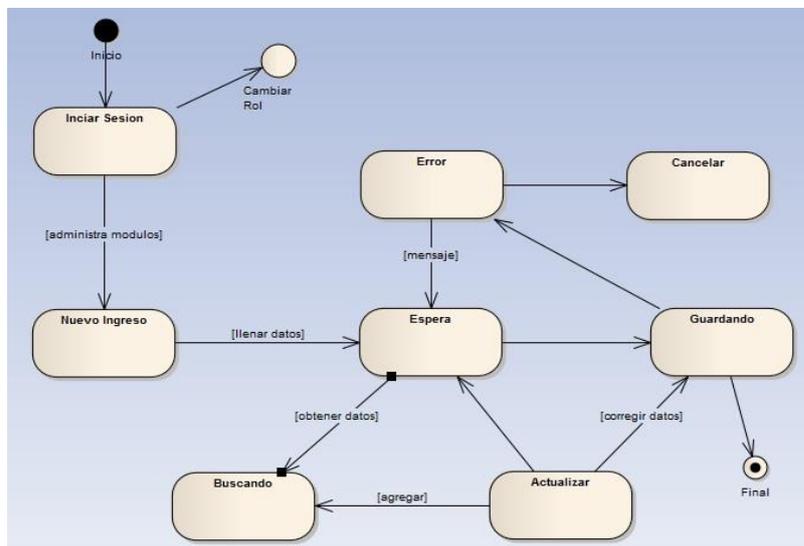


Figura V. 59: Diagrama de transición de estados

Fuente: Autor

Diagramas de Secuencia

Se describe en orden secuencial como el sistema realizar las acciones de cada módulo es decir el ingreso, una consulta, una actualización, utilizando las operaciones, entidades y métodos respectivos, para ello se va a dividir por módulos y entidades:

Módulo Académico: Este módulo realiza acciones académicas como inscripciones, registro de alumno, Año lectivo y matriculas.

Ingreso de un nuevo año lectivo (figuras V.60) donde el administrador cambia de estado los años lectivos de activo a inactivo, para quedar como actual año lectivo el nuevo ingresado.

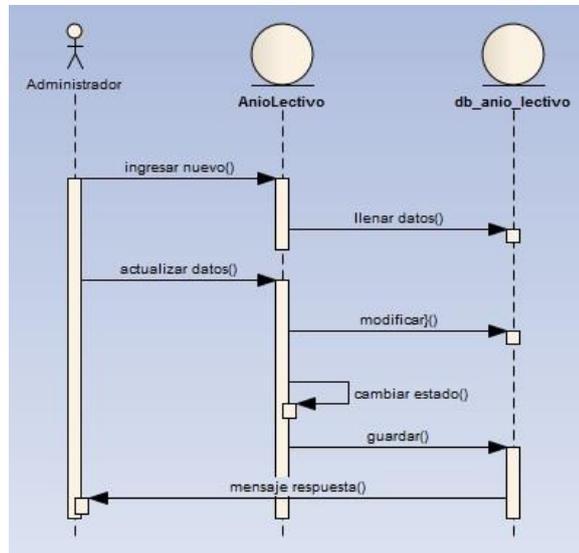


Figura V. 60: Diagrama de secuencia del objeto AnioLectivo

Fuente: Autor

Realiza la inscripción de alumnos (figuras V.61) con el ingreso de datos básicos.

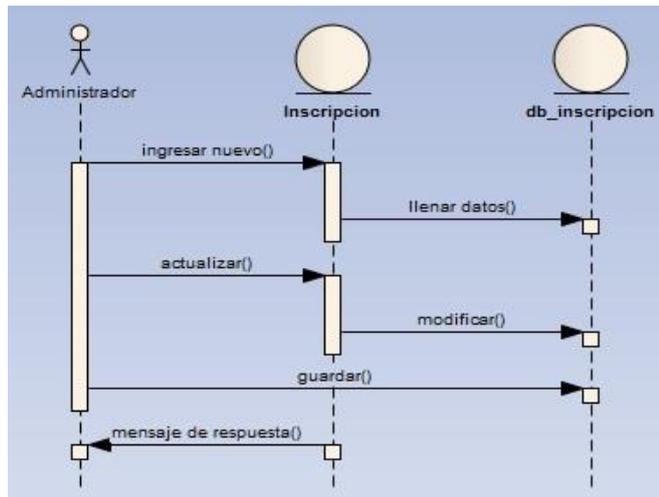


Figura V. 61: Diagrama de Secuencia del objeto inscripción

Fuente: Autor

Posteriormente para realizar la matrícula (figuras V.62) realiza una búsqueda del alumno y lo agrega a nueva matrícula, luego deber llena los datos correspondientes de la matrícula.

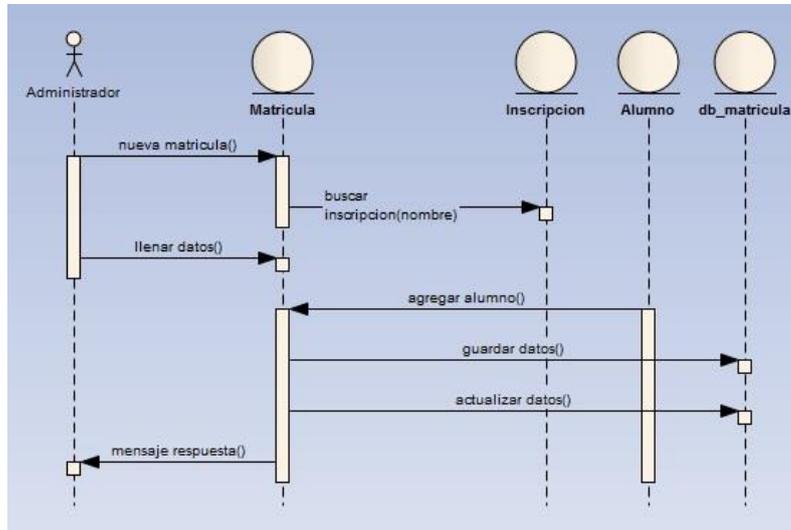


Figura V. 62: Diagrama de secuencia del objeto matrícula

Fuente: Autor

Se realiza el registro de los datos personales del alumno, de los padres de familia y su representante en caso de ser otra persona como se observa en la figura V.63.

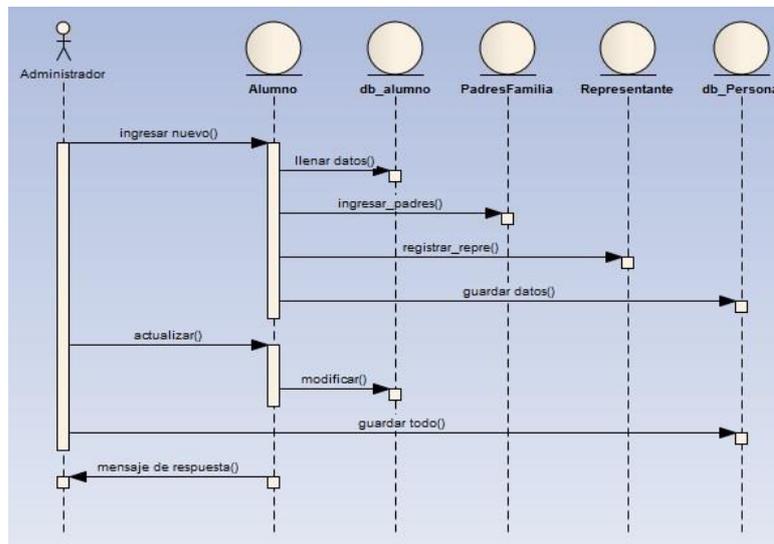


Figura V. 63: Diagrama de secuencia de registro de alumno

Fuente: Autor

Módulo Administración: Este módulo realiza las acciones registro de pago de aranceles y el registro de los datos del maestro.

El registro de pagos (figuras V.64) se lleva a cabo cuando el representante o padre de familia entrega el recibo cancelado en tesorería del arancel correspondiente.

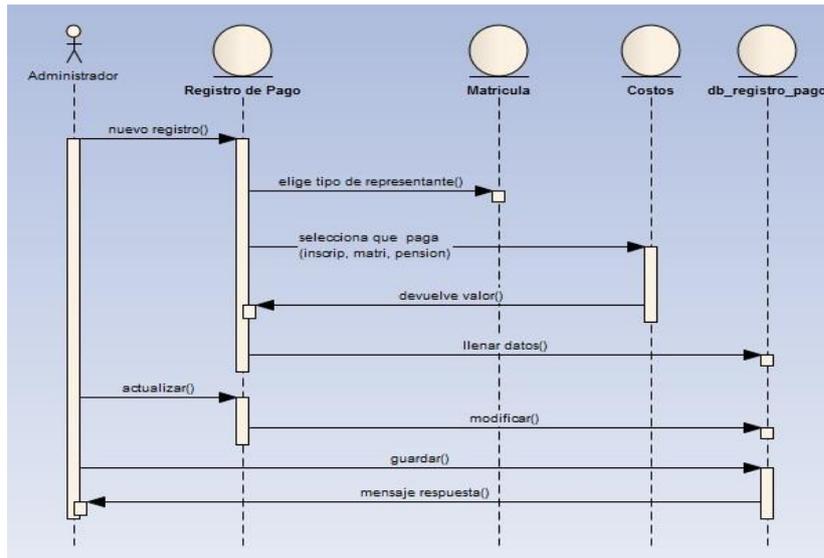


Figura V. 64: Diagrama de secuencia del objeto registro de pago

Fuente: Autor

El registro de maestro consta de un nuevo ingreso a la entidad Maestro, luego llenar los datos y seguidamente guardar en la base de datos, así también se tiene la operación de actualizar,.

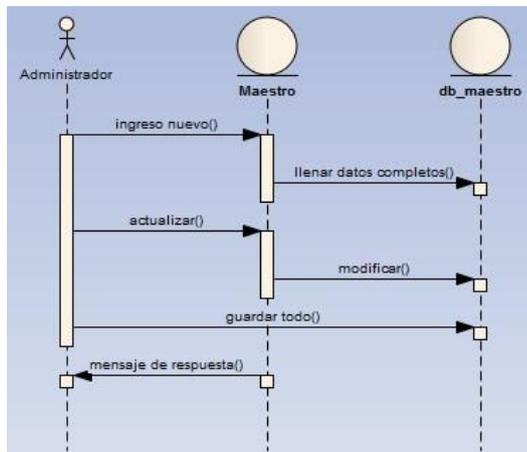


Figura V. 65: Diagrama de secuencia del objeto Maestro

Fuente: Autor

Módulo Catálogos: El administrador tiene la facilidad de controlar el ingreso y actualización de las materias, cursos y paralelos que presenta el centro parvulario, (figuras V.66 a V.69)

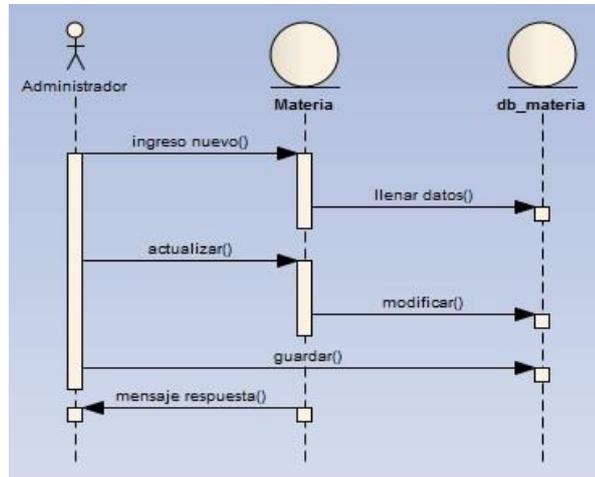


Figura V. 66: Diagrama de secuencia del objeto materia

Fuente: Autor

Estos diagramas de secuencia indican como el usuario administrador selecciona el nuevo ingreso, llena datos y almacena en la base de datos.

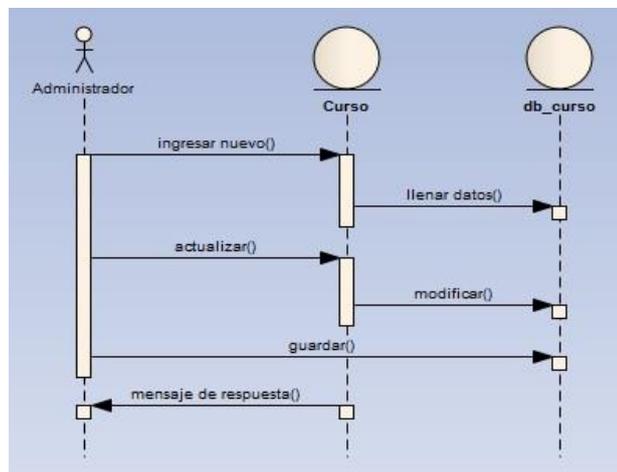


Figura V. 67: Diagrama de secuencia del objeto curso

Fuente: Autor

También para los mismos objetos selecciona actualizar, modifica los datos y guarda en la base de datos.

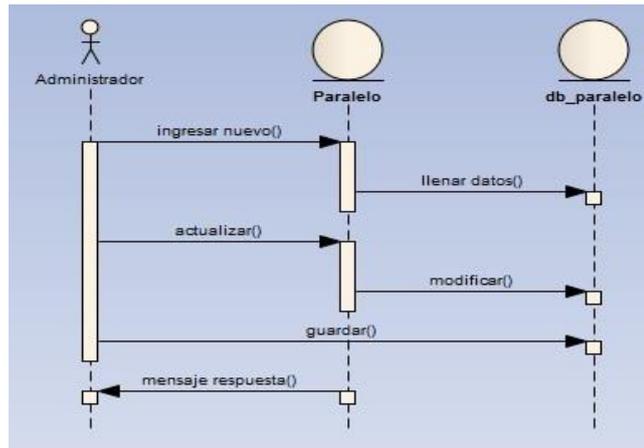


Figura V. 68: Diagrama de secuencia del objeto paralelo

Fuente: Autor

Posteriormente se encuentra la entidad de materia asignada en la cual se puede asignar un grupo de materias a un determinado curso como se observa en la figura V.69.

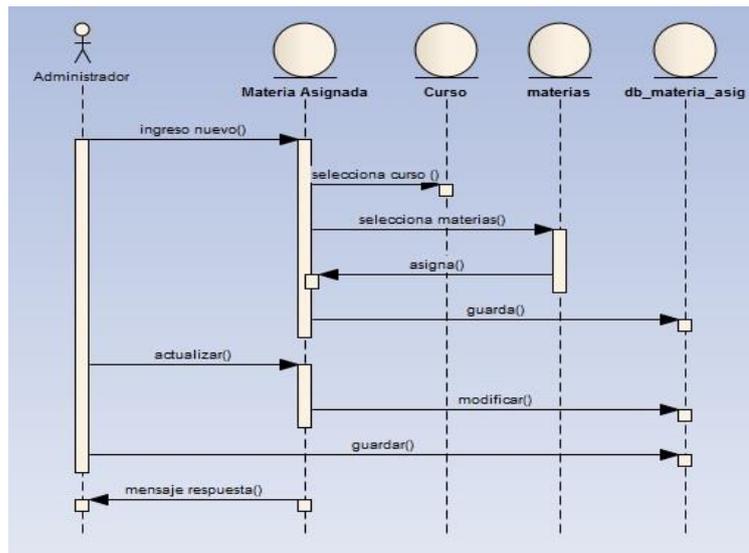


Figura V. 69: Diagrama de secuencia del objeto cuentas de materia asignada

Fuente: Autor

Modulo Reportes: El siguiente diagrama interpreta el proceso que efectúa para la obtención de un reporte tanto para el administrador como para el profesor como se muestra en la figura V.70.

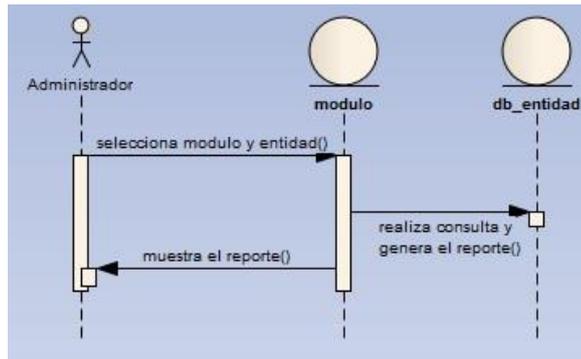


Figura V. 70: Diagrama de secuencia de reportes

Fuente: Autor

PERFIL DE USUARIO MAESTRO

Módulo Evaluación: El maestro realiza las evaluaciones respectivas de cada alumno como se observa en la figura V.71.

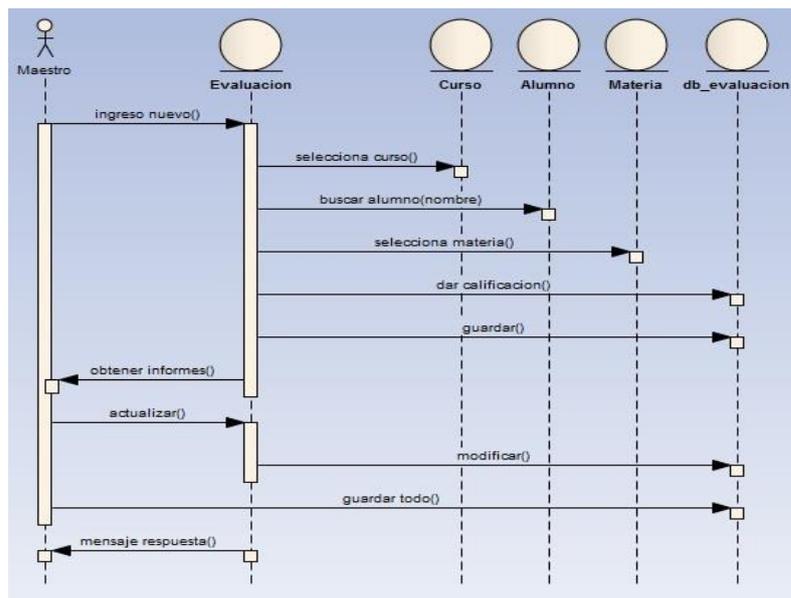


Figura V. 71: Diagrama de Secuencia del objeto evaluación

Fuente: Autor

También permite llevar el control del registro de maestros tanto sus datos personales como los datos académicos (figura V.72).

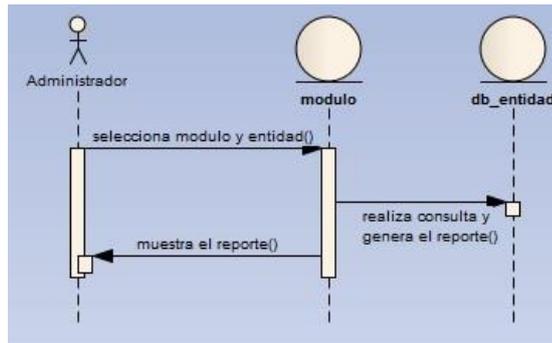


Figura V. 72: Diagrama de secuencia del objeto reporte evaluación

Fuente: Autor

PERFIL ADMINISTRADOR MASTER

MODULO USUARIOS: El siguiente modulo consta de cuatro entidades que van a ser administrados de manera específica por su alta importancia en el manejo, en el cual el administrador puede llevar un control de usuarios con la creación de nuevas cuentas de usuarios (figura V.73).

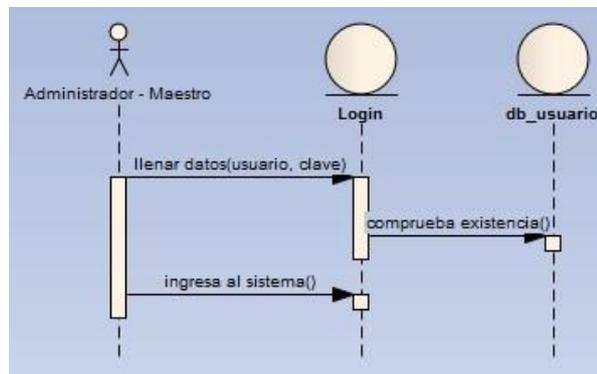


Figura V. 73: Diagrama de secuencia de la obtención de reportes

Fuente: Autor

5.2.1.3. Modelado de Navegación

El siguiente modelado permite establecer los requisitos de navegación por medio de un Diagrama de Usuarios y un Modelo Navegacional.

Mientras que para determinar los requisitos de presentación se desarrolla un Modelo de Presentación.

Diagrama de Usuarios

El diseño de los siguientes diagramas permite identificar el número de perfiles de usuarios que posee el sistema mediante la descripción de las acciones que realiza cada uno de ellos, es decir para este caso el sistema cuenta con dos tipos de perfiles, el administrador (figura V.74) que tiene acceso a todas las acciones.

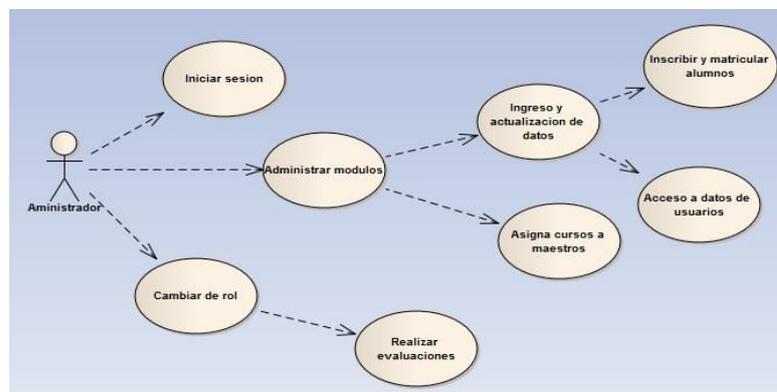


Figura V. 74: Caso de uso del rol administrador

Fuente: Autor

El perfil de usuario o rol de Maestro (Evaluador Parvulario) se representa en el siguiente diagrama de caso de uso (figura V.75) que tiene acceso solo para evaluación de alumnos.

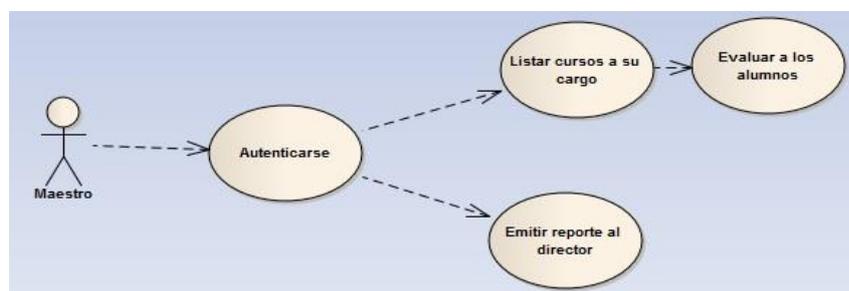


Figura V. 75: Caso de uso del rol maestro

Fuente: Autor

Definición de roles

Teniendo en cuenta las necesidades de cada uno de los usuarios antes mencionados se definió y asignó los roles como se muestra a continuación.

Tabla V. XXXI: Definición de roles

Rol	Cargo	Persona	Descripción
Administrador	Directora del centro parvulario	Lcda. Ximena Pinos	Acceso completo y opción de cambio de rol
Administrador	Secretaria	Lcda. Tania Maldonado	Acceso completo al sistema
Maestro	Profesores	Varios	Acceso al perfil de maestro.
Maestro	Profesora	Lcda. Ximena Pinos	Acceso al perfil de maestro con opción de cambio de rol.

Fuente: Autor

Construcción del Mapa Navegacional: Realiza la descripción de como el usuario puede interactuar con el sistema pero mediante la navegación entre módulos, es decir parte de un menú en el cual se encuentran las entidades de control y realiza las acciones del sistema (figura V.76). Consta de nodos navegacionales que se refiere a las entidades de control y los enlaces permiten la comunicación entre ellos.

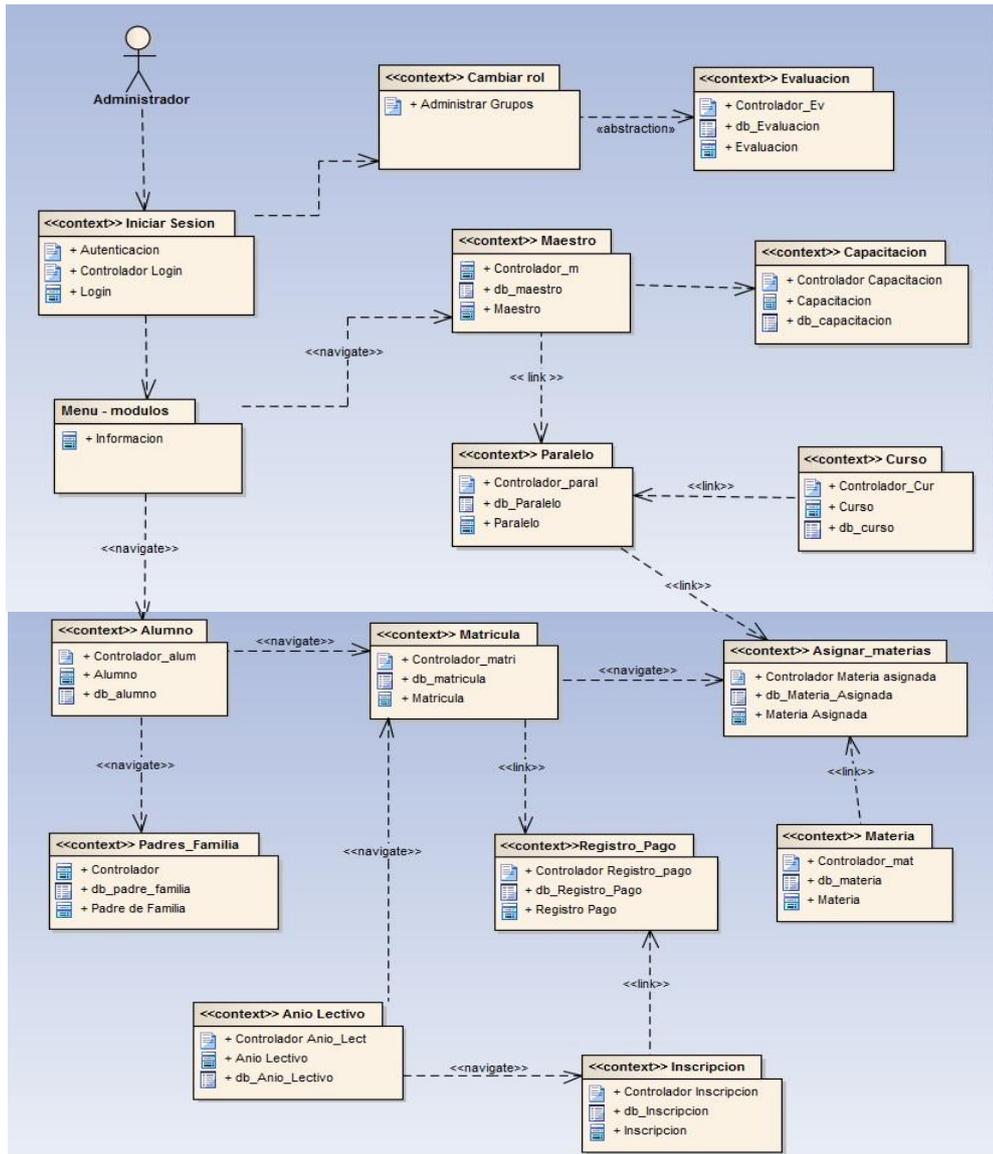


Figura V. 76: Mapa Navegacional

Fuente: Autor

5.2.1.4. Modelo de Presentación

Para el desarrollo de este modelo se diseña el contexto de cada nodo definido en el diagrama de navegacion, especificando sus características como el patron, la cardinalidad, ordenacion y paginacion.

Paginación de información: En el caso del sistema del centro parvulario, vamos a construir bloques de información no más de 10 filas, de todo el conjunto de resultados obtenidos al

momento de realizar consultas, además estos bloques de información contarán con las siguientes características.

- Determinar la página actual en la que el usuario se encuentra.
- Cada conjunto de información no debe estar subrayada.
- Proporcionar vínculos adicionales de Anterior y Siguiente.
- Tener a los vínculos que especifican cada página bajo una distancia relativa.

Ordenación: La información presentada por el sistema, estará ordenada de acuerdo al tipo de información presentada, tomando en cuenta la consulta realizada por el usuario, en casos que la información tenga como relevancia la fecha el reporte estará ordenado de manera ascendente de acuerdo a la fecha, y en los casos en los que los datos presentados por el sistema tengan como relevancia nombres, estarán ordenados de manera ascendente por nombres.

Patrón de presentación: Denominadas relaciones de navegación: El patrón de presentación definirá el modo en que la información de las instancias relacionadas será presentada. Consta de patrones registro y tabular para relaciones “1 a muchos” o de “1 a 1”, respectivamente.

En las siguientes figuras (V.77 – V.89) se muestra la descripción de cada contexto de navegación identificado.

Contexto Alumno: muestra la base de datos, el controlador, los elementos que contiene la vista y, los descriptores de acción: patrón Registro, cardinalidad de n..1, e información Ascendente.

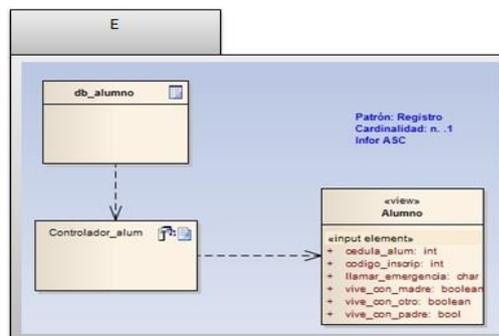


Figura V. 77: Contexto de alumno

Fuente: Autor

Contexto Curso: presenta la base de datos, el controlador, la vista y, descriptores de acción: patron Registro, cardinalidad de 1 . . n, e información Ascendente.

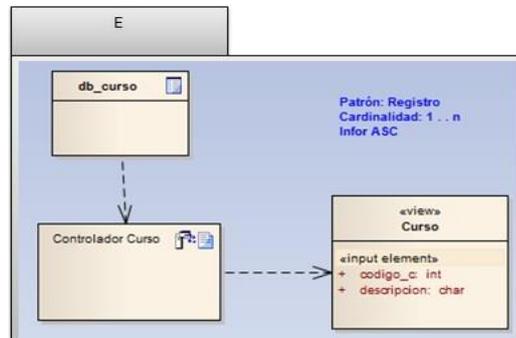


Figura V. 78: Contexto de curso

Fuente: Autor

Contexto Maestro: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de 1 . . n, e información Ascendente.

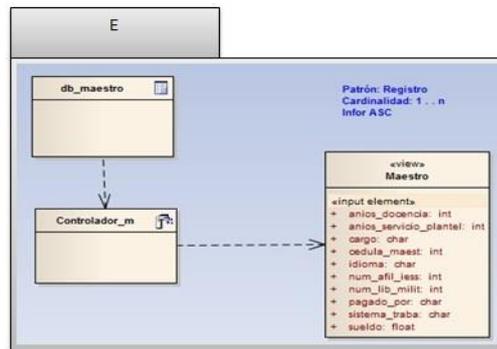


Figura V. 79: Contexto de maestro

Fuente: Autor

Contexto Paralelo: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de n..1, e información Ascendente.

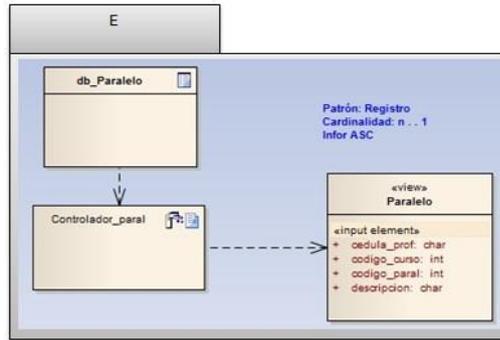


Figura V. 80: Contexto de paralelo

Fuente: Autor

Contexto Padre_familia: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de n..1, e información Ascendente.

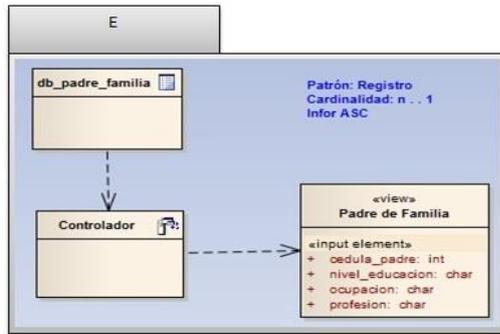


Figura V. 81: Contexto de padres de familia

Fuente: Autor

Contexto Materia_asignada: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de n..1, e información Ascendente.

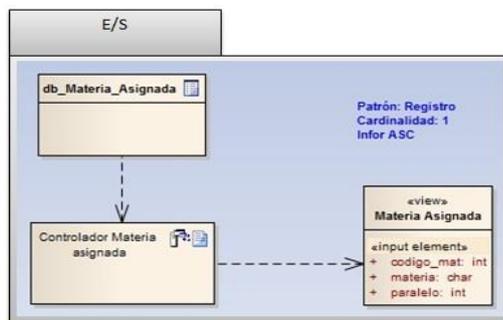


Figura V. 82: Contexto de materia asignada

Fuente: Autor

Contexto Login: presenta el controlador, elementos de la vista y, conexión a la aplicación con descriptores de acción: patron Registro, cardinalidad de n..1, e información Ascendente.

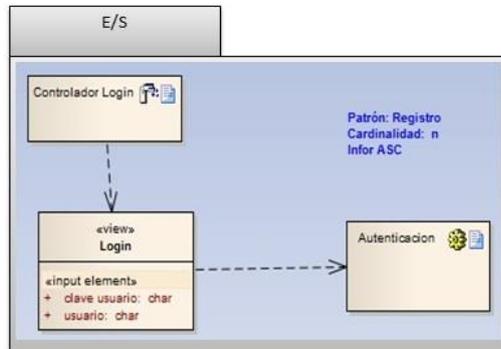


Figura V. 83: Contexto de login o autenticación

Fuente: Autor

Contexto Evaluación: presenta la base de datos, controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de 1..n, e información Ascendente.

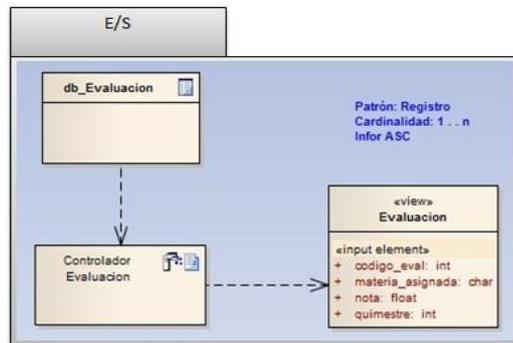


Figura V. 84: Contexto de evaluación

Fuente: Autor

Contexto Matrícula: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de n..1, e información Ascendente.

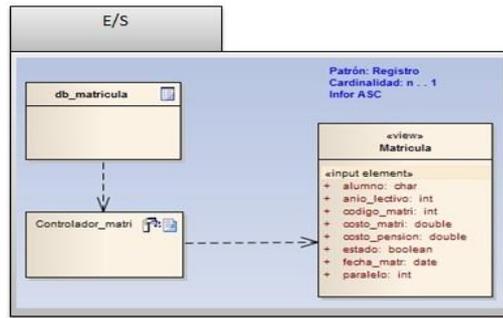


Figura V. 85: Contexto de matrícula

Fuente: Autor

Contexto Registro_Pago: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de n..1, e información Ascendente.

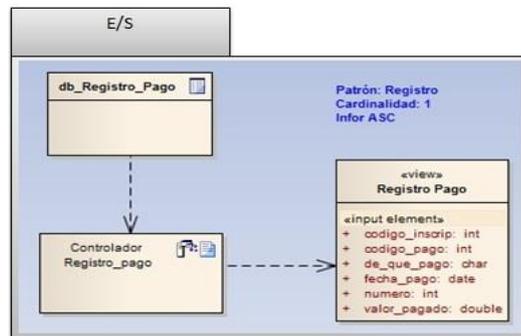


Figura V. 86: Contexto de registro de pago

Fuente: Autor

Contexto Anio_Lectivo: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de 1..n, e información Ascendente.

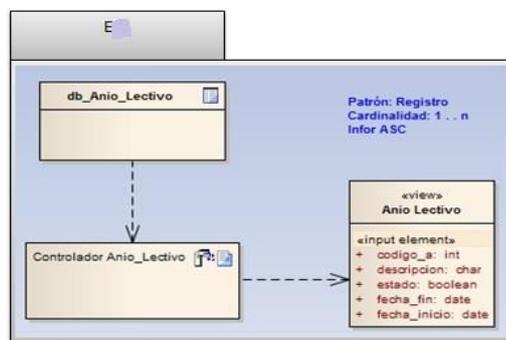


Figura V. 87: Contexto de año lectivo

Fuente: Autor

Contexto Inscripción: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de n..1, e información Ascendente.

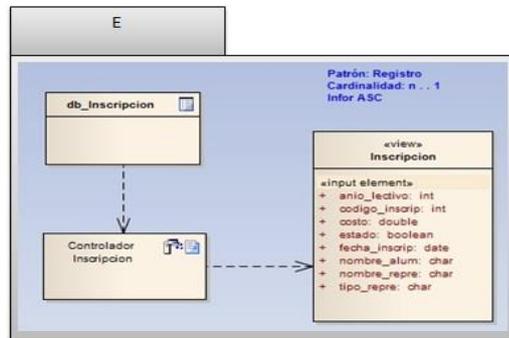


Figura V. 88: Contexto de inscripción

Fuente: Autor

Contexto Materia: presenta la base de datos, el controlador, elementos de la vista y, descriptores de acción: patron Registro, cardinalidad de n..1, e información Ascendente.

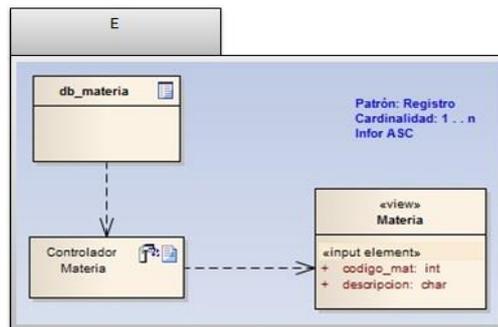


Figura V. 89: Contexto de materia

Fuente: Autor

5.2.2. DESARROLLO DE LA SOLUCION

Esta etapa se desarrolla en 3 niveles o capas que incluye la definición del framework y el servidor web, la codificación y el diseño de base de datos

5.2.2.1. Capa de Presentación

Detalla como la información debe ser presentada a los usuarios, la misma que está conformada por el navegador y el servidor web. El navegador contiene paginas HTML, programas

interpretes (JavaScript) y programas compilados (Applets de java), para facilitar la validación de los datos, ayudar en la navegación y animar la interfaz del usuario.

Desarrollo de Vistas en lenguaje HTML: consta del diseño de interfaces que van a interactuar con el usuario como se muestra en la figura V.90.

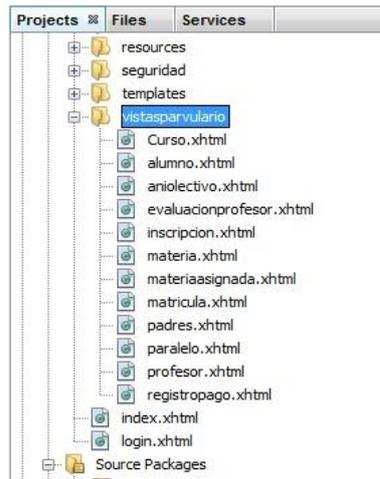


Figura V. 90: Vistas del sistema

Fuente: Autor

Principales Interfaces del Sistema: Se establece como interfaces principales a las siguientes pantallas que son: inicio de sesion (figura V.91), cambio de rol de usuario y la generacion de un reporte.



Figura II. 91: Pantalla final de inicio de sesión

Fuente: Autor

La siguiente pantalla identifica el cambio de rol de usuario (figura V.92), cuando el usuario identificado tiene permisos de cambio ya sea como Administrador Master o Administrador Parvulario.



Figura V. 92: Pantalla final de cambio de rol

Fuente: Autor

Se puede mencionar como una interfaz importante la generación de un reporte (figura V.93), el cual el usuario debe seleccionar el icono PDF del módulo o entidad que requiera.

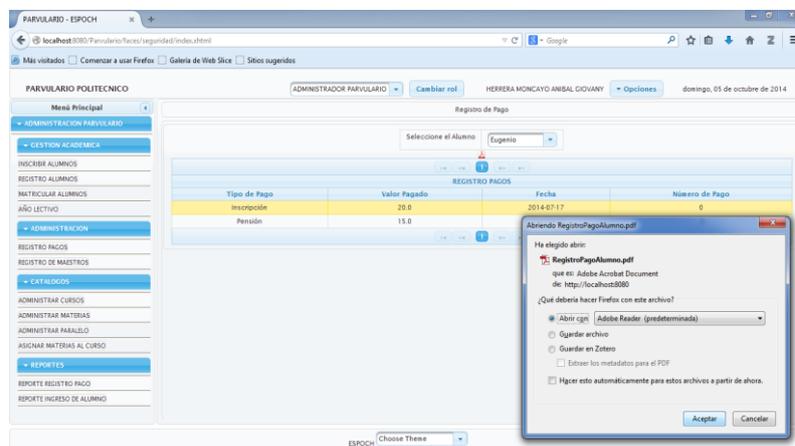


Figura V. 93: Pantalla final de generación de reporte

Fuente: Autor

Lista de mensajes y operaciones del sistema

El sistema cuenta con diferentes tipos de mensajes de acuerdo a la ejecución de una operación, el cual ayudan al usuario a identificar el estado de la transacción.

Mensajes de Error

Debe ingresar el nombre de usuario

Debe ingresar la contraseña

Usuario y/o contraseña incorrectos

Cuando los campos son obligatorios muestra ERROR

Mensajes de Advertencia

Seleccione el nivel de educación

Ingrese la cédula

No insertado

Mensajes Informativos

Inserción Exitosa

Actualización exitosa

Persona Matriculada

No se encontraron datos

Los datos fueron eliminados

OPERACIONES:

- a. Inserción
- b. b. Actualización
- c. c. Edición
- d. d. Impresión

Diseño de Reportes: El diseño de las vistas de presentación de los reportes implementados se realiza en la herramienta JasperReports, el mismo que muestra en la figura V.94.

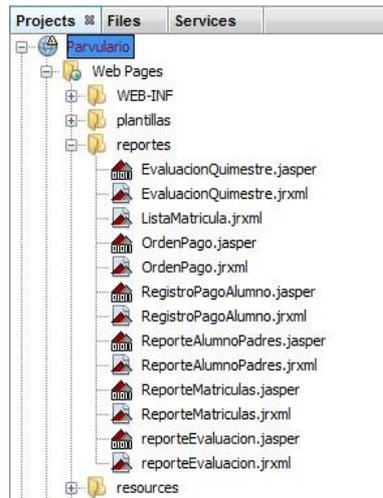


Figura V. 94: Vistas de Presentación de Reportes

Autor: Fuente

Librerías: Para la implementación de realizo el uso de diferentes librerías, para la conexión con la base de datos, implemetación de vistas, diseño de reportes, ejecución del servidor Glass Fish.

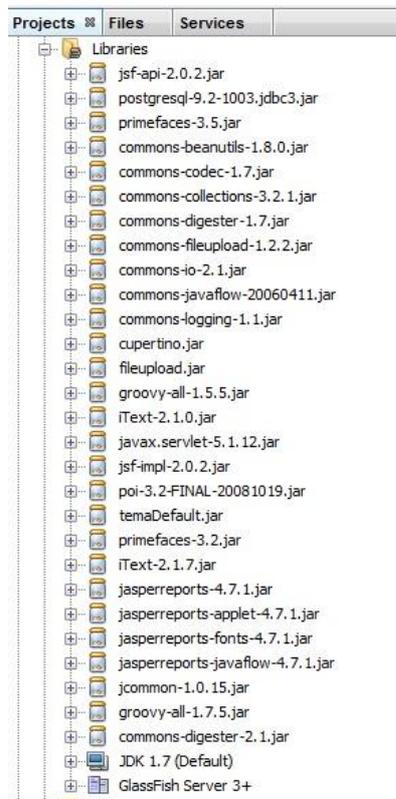


Figura V. 95: Libresrias utilizadas para la implemetación

Fuente: Autor

5.2.2.2. Capa de Aplicación

El IDE de NetBeans permite la implementación de diferentes frameworks que facilitan el buen desarrollo de las aplicaciones, en este caso JSF es un Framework que realiza los módulos CRUD de una forma sumamente sencilla, simplemente con el seguimiento de unos cuantos pasos que facilitan los asistentes de NetBeans. Esta característica hace que las aplicaciones se desarrollen de una forma fácil y adecuada para situaciones en las que el cliente requiere la aplicación rápidamente o que simplemente requiere el desarrollo de un módulo de la aplicación que tenga que ver con altas y bajas.

También se utiliza el servidor **GlassFish3.2.1** de software libre que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

Desarrollo de clases: Se realiza la creación de las clases identificadas en el modelo conceptual, las mismas que se muestran en la figura V.96.

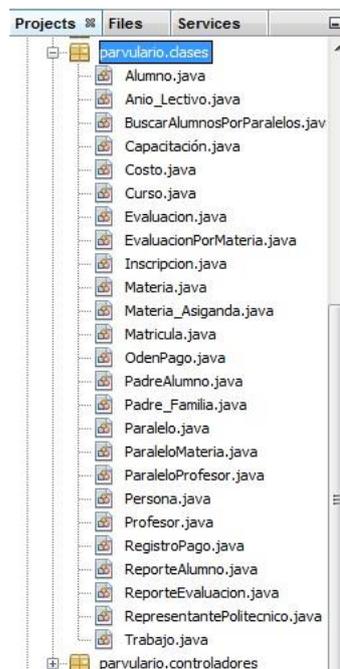


Figura II. 96: Desarrollo de clases del sistema

Fuente: Autor

Desarrollo de Funciones: En la figura V.97, se observa las funciones creadas en la aplicación, donde constan las operaciones principales de cada clase como son insertar, buscar, listar, actualizar y eliminar.

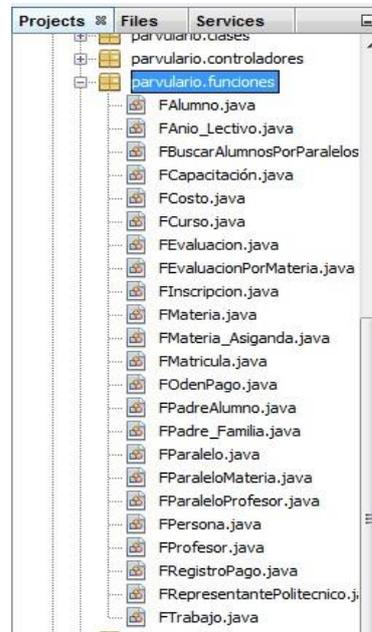


Figura V. 97: Desarrollo de funciones del sistema

Fuente: Autor

Desarrollo de Controladores: Los controladores (figura V.98) manejan la funcionalidad involucrada desde que el usuario entra a la web hasta que este sale.

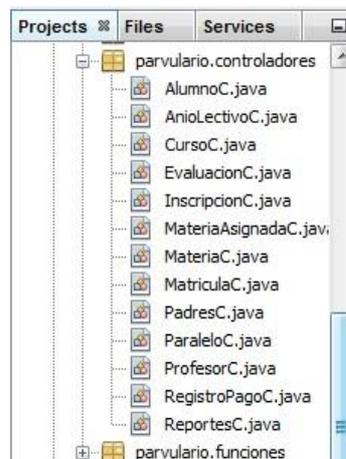


Figura V. 98: Desarrollo de controladores del sistema

Fuente: Autor

Cada clic que accione significa que todo pasa por este componente, es decir llama a objetos de la lógica de negocio del modelo para que resuelvan lo necesario, y según lo que suceda ejecutará la página JSP para mostrarle una salida al usuario.

5.2.2.3. Capa de Base de Datos

En esta capa se especifica el modelado de la base de datos con sus respectivas tablas y relaciones como se observa en la figura V.99.

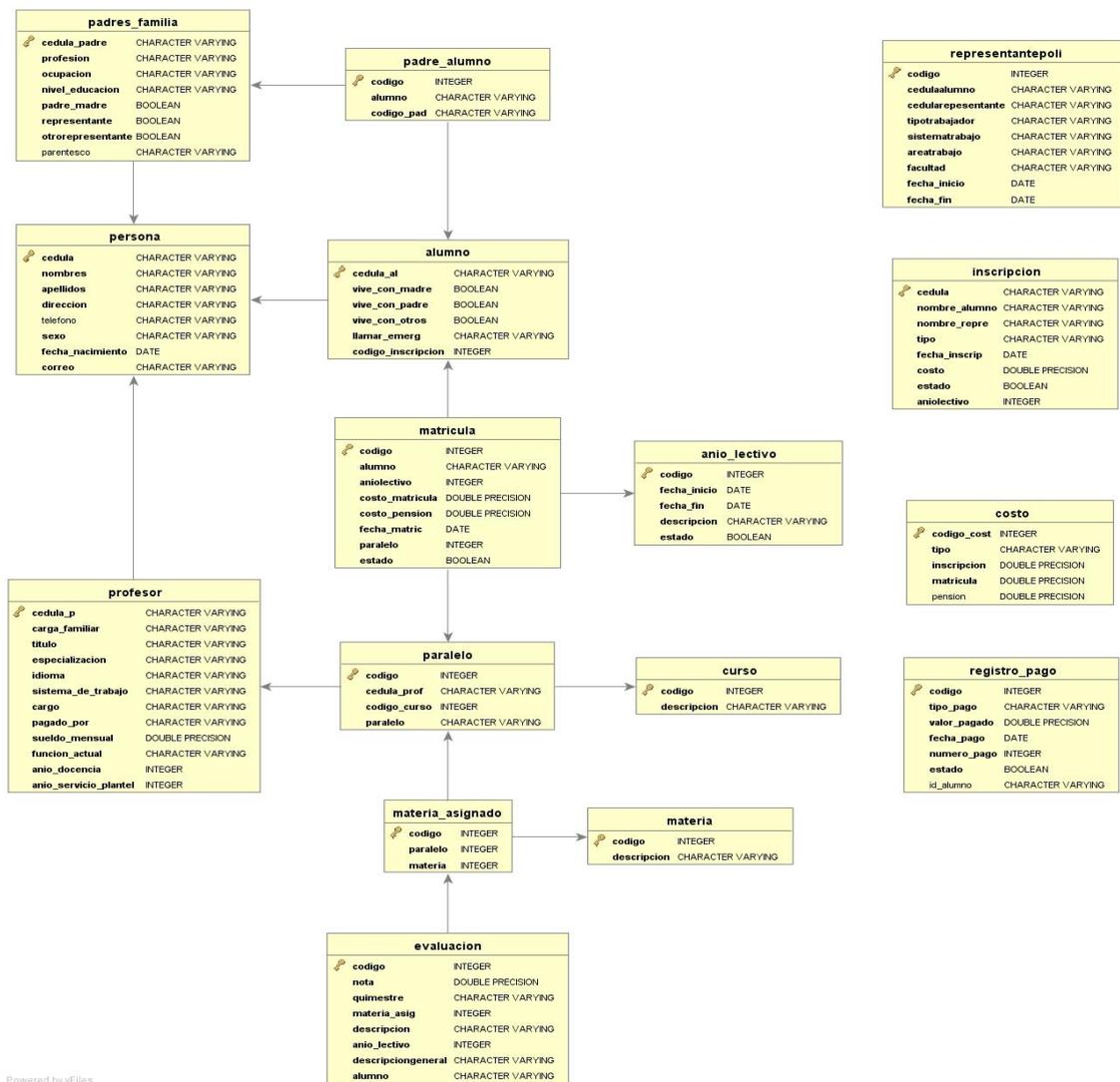


Figura V. 99: Diseño de Base de Datos

Fuente: Autor

Creación de Esquema Parvulario

El esquema de la aplicación se puede crear de forma gráfica o mediante un script, en donde se va almacenar todos los componentes de nuestra base de datos como son tablas, funciones, triggers, secuencias y más, en la figura V.100, se puede observar el esquema completo.

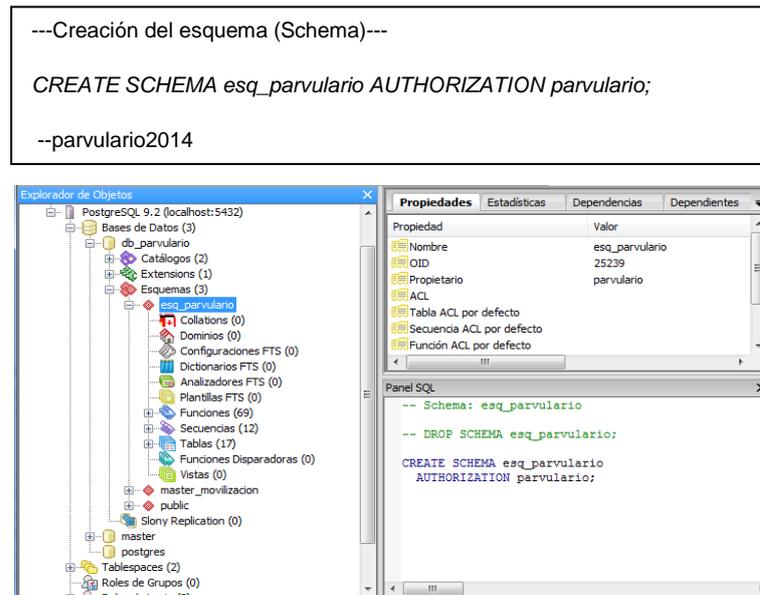


Figura V. 100: Esquema Parvulario

Fuente: Autor

Diccionario de Datos

Es importante especificar los tipos de datos que se utiliza en la base de datos para correcciones futuras por técnicos ajenos a un proyecto. A continuación se describe cada una de las tablas creadas, con sus respectivos atributos.

Tabla anio_Lectivo: registro del periodo académico con un control de estado.

Tabla V. XXXII: Descripción de datos de la tabla anio_lectivo

Nombre de la columna	Tipo de dato	Primary key	Not Null
código	integer	SI	SI
descripcion	character varying	NO	SI

estado	boolean	NO	SI
Fecha_fin	date	NO	SI
Fecha_inicio	date	NO	SI
Sequencia	nextval('esq_parvulario.anio_lectivo_codigo_se q':::regclass)		

Fuente: Autor

Tabla V. XXXIII: Restricción de la tabla anio_lectivo

Nombre	Tipo	Definición
pk_codigo_anio_lectivo	Clave primaria	(codigo)

Fuente: Autor

Tabla alumno: Registro de datos completos de alumno que va a ser matriculado en el centro infantil.

Tabla V. XXXIV: Descripción de datos de la tabla alumno

Nombre de la columna	Tipo de dato	Not Null	Primary Key
Cedula_al	character varying	SI	SI
Vive_con_madre	boolean	SI	NO
Vive_con_padre	boolean	SI	NO
Vive_con_otros	boolean	SI	NO
Llamar_emerg	character varying	SI	NO
Código_inscrip	integer	SI	NO

Fuente: Autor

Tabla V. XXXV: Restricción de la tabla alumno

Nombre	Tipo	Definición
pk_codigo_alun	Clave primaria	(cedula_al)
fk_codigo_person a_al	Clave ajena	(cedula_al) REFERENCES esq_parvulario.persona (cedula)

Fuente: Autor

Tabla Inscripción: Contiene los datos básicos del alumno para su registro.

Tabla V. XXXVI: Descripción de datos de la tabla inscripción

Nombre de la columna	Tipo de dato	Not Null	Primary Key
anio_lectivo	integer	SI	NO
Código_inscrip	integer	SI	SI
costo	double precision	SI	NO
estado	boolean	SI	NO
Fecha_inscrip	date	SI	NO
Nombre_alumno	character varying	SI	NO
Nombre_repre	character varying	SI	NO
Tipo_repre	character varying	SI	NO
Secuencia:	nextval('esq_parvulario.inscripcion_codigo_inscrip_seq':regclass)		

Fuente: Autor

Tabla V. XXXVII: Restricciones de la tabla inscripción

Nombre	Tipo	Definición
pk_inscrip	Clave primaria	(codigo_inscrip)

fk_anio_lectivo	Clave ajena	(aniolectivo) esq_parvulario.anio_lectivo (codigo)	REFERENCES
-----------------	-------------	---	------------

Fuente: Autor

Tabla Matricula: La siguiente tabla guarda datos específicos de matrícula pero se relaciona con el alumno, curso y paralelo.

Tabla V. XXXVIII: Descripción de datos de la tabla matrícula

Nombre de la columna	Tipo de dato	Not Null	Primary key
codigo	integer	SI	SI
anio_lectivo	integer	SI	NO
alumno	character varying	SI	NO
costo_matri	double precision	SI	NO
costo_pension	double precision	SI	NO
estado	boolean	SI	NO
fecha_matri	date	SI	NO
paralelo	integer	SI	NO
Secuencia:	nextval('esq_parvulario.matricula_codigo_seq'::regclass)		

Fuente: Autor

Tabla V. XXXIX: Restricciones de tabla matricula

Nombre	Tipo	Definición
pk_codigo_matricula	Clave primaria	(codigo)
fk_codigo_alumno_m	Clave ajena	(alumno) REFERENCES esq_parvulario.alumno (cedula_al)

fk_codigo_aniolectivo_m	Clave ajena	(aniolectivo) REFERENCES esq_parvulario.anio_lectivo (codigo)
fk_codigo_paralelo_m	Clave ajena	(paralelo) REFERENCES esq_parvulario.paralelo (codigo)

Fuente: Autor

Tabla persona: Esta tabla almacena datos correspondientes a una persona, la cual relaciona a todas las tablas de tipo persona para completar su información.

Tabla V. XL: Descripción de datos de tabla persona

Nombre de la columna	Tipo de dato	Not Null	Primary key
cedula	character varying	SI	SI
nombres	character varying	SI	NO
apellidos	character varying	SI	NO
fecha_nac	date	SI	NO
direccion	character varying	SI	NO
telefono	character varying	NO	NO
sexo	character varying	SI	NO
correo	character varying	SI	NO

Fuente: Autor

Tabla V. XLI: Restricción de tabla persona

Nombre	Tipo	Definición
pk_codigo_persona	Clave primaria	(cedula)

Fuente: Autor

Tabla padres: Almacena datos completos tanto del padre como de la madre y se relaciona con el alumno.

Tabla V. XLII: Descripción de datos de la tabla padre

Nombre de la columna	Tipo de dato	Not Null	Primary key
cedula_padre	character varying	SI	SI
nivel_educcion	character varying	SI	NO
ocupacion	character varying	SI	NO
profesion	character varying	SI	NO
Padre_madre	boolean	SI	NO

Fuente: Autor

Tabla V. XLIII: Restricciones de tabla padres_familia

Nombre	Tipo	Definición
pk_codigo_padres	Clave primaria	(cedula_padre)
fk_codigo_padre	Clave ajena	(cedula_padre) REFERENCES esq_parvulario.persona (cedula)

Fuente: Autor

Tabla padre_alumno: Esta tabla es la relación de n...n entre la tabla alumno y padre de familia.

Tabla V. XLIV: Descripción de datos de la tabla padre_alumno

Nombre de la columna	Tipo de dato	Primary key	Not Null
código	integer	SI	SI
alumno	character varying	SI	NO
codigo_pad	character varying	SI	NO
Secuencia:	nextval('esq_parvulario.padre_alumno_codigo_s		

	eq':::regclass)
--	-----------------

Fuente: Autor

Tabla V. XLV: Restricciones de tabla padre_alumno

Nombre	Tipo	Definición
pk_codigo_p_a	Clave primaria	(codigo)
fk_codigo_padre_alu mno_p_a	Clave ajena	(alumno) REFERENCES esq_parvulario.alumno (cedula_al)
fk_codigo_padre_p_ a	Clave ajena	(codigo_pad) REFERENCES esq_parvulario.padres_fa milia (cedula_padre)

Fuente: Autor

Tabla representante: Almacena los datos básicos de la persona desinada como representante del alumno.

Tabla V. XLVI: Descripción de datos de la tabla representante

Nombre de la columna	Tipo de dato	Not Null	Primary key
Cedula_repre	character varying	SI	SI
tipo	character varying	SI	NO
Estado_civil	character varying	SI	NO

Fuente: Autor

Tabla V. XLVII: Restricciones de tabla representante

Nombre	Tipo	Definición
pk_represen	Clave primaria	(cedula_repre)

fk_represent	Clave ajena	(cedula_repre) REFERENCES esq_parvulario.persona (cedula)
--------------	-------------	--

Fuente: Autor

Tabla profesor: Se almacena los datos personales y académicos del profesor.

Tabla V. XLVIII: Descripción de datos de la tabla profesor

Nombre de la columna	Tipo de dato	Not Null	Primary key
cedula_p	character varying	SI	SI
Carga_familiar	character varying	SI	NO
titulo	character varying	SI	NO
especializacion	character varying	SI	NO
idioma	character varying	SI	NO
sistema_de_trabajo	character varying	SI	NO
cargo	character varying	SI	NO
Pagado_por	character varying	SI	NO
Sueldo_mensual	Double precision	SI	NO
Función_actual	character varying	SI	NO
Anio_docencia	Integer	SI	NO
Anio_servicio_plantel	integer	SI	NO

Fuente: Autor

Tabla V. XLIX: Restricciones de tabla profesor

Nombre	Tipo	Definición
pk_codigo_profesor	Clave primaria	(cedula_p)
fk_codigo_persona	Clave ajena	(cedula_p) REFERENCES

		esq_parvulario.persona (cedula)
--	--	------------------------------------

Fuente: Autor

Tabla Representantepoli: En esta tabla almacena los datos del tipo de representate.

Tabla V. L: Descripción de datos de la tabla representantepoli

Nombre de la columna	Tipo de dato	Not Null	Primary key
código	serial	SI	SI
Cedulaalumno	varchar	SI	NO
tipotrabajador	varchar	SI	NO
sistematrabajo	varchar	SI	NO
areatrabajo	varchar	SI	NO
facultad	varchar	SI	NO
Fecha_inicio	varchar	SI	NO
Fecha_fin	date	NO	NO

Fuente: Autor

Tabla V. LI: Restricciones de la tabla representantepoli

Nombre	Tipo	Definición
pk_codigo	Clave primaria	(codigo)

Fuente: Autor

Tabla curso: En la siguiente tabla guarda el nombre de los cursos que tiene la institución mediante el campo descripción.

Tabla V. LII: Descripción de datos de la tabla curso

Nombre de la columna	Tipo de dato	Not Null	Primary key
codigo	integer	SI	SI
descripcion	character varying	SI	NO
Secuencia:	nextval('esq_parvulario.curso_codigo_seq':reg class)		

Fuente: Autor

Tabla V. LIII: Restricción de la tabla curso

Nombre	Tipo	Definición
pk_codigo_curso	Clave primaria	(codigo)

Fuente: Autor

Tabla paralelo: Al igual que la tabla anterior solo almacena los paralelos representados en letras mayúsculas, pero están relacionados con las tablas curso y profesor.

Tabla V. LIV: Descripción de datos de la tabla paralelo

Nombre de la columna	Tipo de dato	Not Null	Primary key
código	integer	SI	SI
paralelo	character varying	SI	NO
cedula_prof	character varying	SI	NO
código_curso	integer	SI	NO

Fuente: Autor

Tabla V. LV: Restricciones de la tabla paralelo

Nombre	Tipo	Definición
pk_codigo_paralelo	Clave primaria	(codigo)
fk_codigo_curso	Clave ajena	(codigo_curso) REFERENCES esq_parvulario.curso (codigo)
fk_codigo_curso_dos	Clave ajena	(cedula_prof) REFERENCES esq_parvulario.profesor (cedula_p)

Fuente: Autor

Tabla materias: En esta tabla guarda los nombres de las materias que se imparte en cada curso.

Tabla V. LVI: Descripción de datos de la tabla materia

Nombre de la columna	Tipo de dato	Not Null	Primary key
código	integer	SI	SI
descripcion	character varying	SI	NO
Secuencia:	nextval('esq_parvulario.materia_codigo_seq'::regclass)		

Fuente: Autor

Tabla V. LVII: Restricción de tabla materia

Nombre	Tipo	Definición
pk_codigo_materia	Clave primaria	(codigo)

Fuente: Autor

Tabla materia_asignada: La siguiente tabla se relaciona con paralelo, materia y evaluación por considerarse necesario la asignación de materias a cada curso de acuerdo al paralelo y para llevar el control de evaluación.

Tabla V. LVIII: Descripción de datos de la tabla materia asignada

Nombre de la columna	Tipo de dato	Primary key	Not Null
código	integer	SI	SI
materia	integer	SI	NO
paralelo	integer	SI	NO
Secuencia:	nextval('esq_parvulario.materia_asignado_codigo_seq'::regclass)		

Fuente: Autor

Tabla V. LIX: Restricciones de la tabla materia_asignada

Nombre	Tipo	Definición
pk_codigo_materia_asignado	Clave primaria	(codigo)
fk_codigo_materia	Clave ajena	(materia) REFERENCES esq_parvulario.materia (codigo)
fk_codigo_paralelo_materia	Clave ajena	(paralelo) REFERENCES esq_parvulario.paralelo (codigo)

Fuente: Autor

Tabla evaluación: Los datos para esta tabla son cortos y específicos para la evaluación del alumno por materia y curso, y está relacionada con la tabla materia asignada.

Tabla V. LX: Descripción de datos de la tabla evaluación

Nombre de la columna	Tipo de dato	Not Null	Primary key
código	integer	SI	SI
descripcion	character varying	SI	NO
materia_asig	integer	SI	NO
nota	double precision	SI	NO
quimestre	character varying	SI	NO
Secuencia:	nextval('esq_parvulario.evaluacion_codigo_seq'::regclass)		

Fuente: Autor

Tabla V. LXI: Restricciones de la tabla evaluación

Nombre	Tipo	Definición
pk_codigo_evaluacion	Clave primaria	(codigo)
fk_codigo_evaluacion	Clave ajena	(materia_asig) REFERENCES esq_parvulario.materia_asignado (codigo)

Fuente: Autor

Tabla registro_pago: Esta tabla no está relacionada con ninguna otra tabla a nivel de base de datos, ya que solo almacena los pagos de los aranceles realizados en la tesorería de la ESPOCH.

Tabla V. LXII: Descripción de datos de la tabla registro pago

Nombre de la columna	Tipo de dato	Not Null	Primary key
código	integer	SI	SI
matricula	integer	SI	NO

fecha_pago	date	SI	NO
Valor_pago	double precision	SI	NO
Estado_matricula	character varying	SI	NO
Numero_pago	integer	SI	NO
estado	boolean	SI	NO
Secuencia:	nextval('esq_parvulario.registro_pago_codigo_seq'::regclass)		

Fuente: Autor

Tabla V. LXIII: Restricción de la tabla registro_pago

Nombre	Tipo	Definición
pk_codigo_registro_pago	Clave primaria	(codigo)

Fuente: Autor

Tabla costo: Esta tabla almacena los valores correspondiente de pago por lo aranceles del centro infantil, y a nivel de base de datos tampoco tiene ninguna relación.

Tabla V. LXIV: Descripción de datos de la tabla costo

Nombre de la columna	Tipo de dato	Primary key	Not Null
código_cost	integer	SI	SI
inscripcion	double precision	SI	NO
matricula	double precision	SI	NO
pension	double precision	SI	NO
tipo	character varying	NO	NO
Secuencia:	nextval('esq_parvulario.costo_codigo_cost_seq'::regclass)		

Fuente: Autor

Tabla V. LXV: Restricción de la tabla costo

Nombre	Tipo	Definición
pk_codigo_registro_pago	Clave primaria	(codigo)

Fuente: Autor

Con la finalización de la capa de base de datos se llega a la culminación de la aplicación para el Parvulario Politecnico, cabe mencionar que para el desarrollo del proyecto se empleó la reutilización de código, tomando el módulo de Gestión de usuarios del Esquema Master_movilizacion presentado en el anexo 6.

CONCLUSIONES

- El estudio realizado de las metodologías web tanto de UWE y OOWS demuestran que ambas pretenden un mismo objetivo, desarrollar métodos de producción de software que permitan construir aplicaciones web complejas con comportamiento dinámico, que sean compatibles con los estándares metodológicos y de notación más extendidos en la actualidad.
- Para lograr una mayor efectividad y veracidad en la comparación de las metodologías en cuestión, fue de vital importancia la realización de prototipos siguiendo cada uno de los pasos del proceso de desarrollo, y estableciendo indicadores y métricas apropiadas para la evaluación final.
- La utilización de herramientas de tipo Open Source permiten reducir costos finales del sistema sin dejar a un lado la calidad del producto, logrando una mayor comprensión y fácil aplicabilidad para el código programado.
- Mediante la valoración porcentual de los resultados de la evaluación se llega a la conclusión que la metodología UWE obtuvo el 63,86%, mientras que la metodología OOWS obtuvo el 84,22%, demostrando la hipótesis propuesta.
- La investigación realizada nos lleva a demostrar que el utilizar una metodología no tan común pero óptima en el desarrollo como lo es OOWS permite ganar tiempo al programador y centrarse en objetivos específicos que una aplicación web debe tener.

RECOMENDACIONES

- De acuerdo al estudio realizado es importante que desarrollador tome en cuenta hacia donde se dirige el uso de la aplicación web, para una correcta selección de la metodología.
- También es recomendable guiarse en estudios realizados de diferentes metodologías, que nos ayuden a descubrir una opción aceptable y adecuada para la producción de software en un alto índice en productividad.
- La determinación de indicadores deben estar basados en algún modelo, para que tengan una validez efectiva a la hora de la evaluación.
- Para garantizar una evaluación apropiada es recomendable analizar los datos sistemáticamente y de acuerdo a los indicadores propuestos, mediante la utilización de métodos matemáticos y estadísticos.
- Es importante tener en cuenta que el sistema es una herramienta que contribuirá para la toma de decisiones de la institución, por tanto los datos a ser ingresados en las pruebas del sistema deben ser lo más cercanos a la realidad.
- La metodología OOWS incluye un paso no desarrollado, por lo que se recomienda que debe ser explorado en un futuro, la Implementación automática de código que requiere desarrollar un capítulo de investigación sobre ese proceso.

RESUMEN

El propósito de la investigación es realizar un estudio comparativo entre las metodologías UWE y OOWS para mejorar la productividad en el desarrollo de aplicaciones web y, con la metodología seleccionada, implementar un sistema para aplicar en el parvulario de la ESPOCH, con el propósito de automatizar los procesos académicos.

Se utilizó diferentes métodos de investigación como el método científico en el planteamiento del problema, levantamiento de información, análisis e interpretación de resultados y, en la comprobación de hipótesis, el método experimental para la elaboración de un prototipo por cada metodología, para el desarrollo del módulo “Matricula” se utilizó herramientas software como Java Prime Face, PostgreSQL, Enterprise Architec.

Se obtuvo los siguientes resultados para los indicadores de: Evaluación en base al tipo de requisitos, UWE alcanzó 66, 67% y OOWS el 100%; Evaluación en base a la propuesta funcional, UWE obtuvo 33,33% mientras que OOWS logró 66,67%; Evaluación del tiempo de diseño, UWE alcanzó el 61,82% mientras que OOWS obtuvo el 80,87%; Evaluación del modelo de producto: UWE con 78,79% y OOWS obtuvo el 93,94%; Evaluación del modelo de proceso: UWE obtuvo el 75,76% mientras OOWS alcanzó el 90,91%; Evaluación por puntos de función obtuvo UWE: 66,77% y OOWS: 73,10%. Con la aplicación de estadística descriptiva se logró el resultado final, seleccionándose la metodología OOWS al obtener el mayor puntaje en productividad, 84,22%.

La implementación de la aplicación web desarrollado en el Parvulario Politécnico logró el objetivo de mejorar la productividad.

SUMMARY

The purpose of the research is to carry out a comparative study between UWE and OOWS methodologies to improve productivity in the development of web applications with the selected methodology, implementing a system to be applied in the kindergarten of ESPOCH, in order to automate academic processes.

Different methods of investigation such as the Scientific Method in the problem statement, information gathering, analysis and interpretation of results and in the hypothesis testing the Experimental Method for the production of a prototype for each methodology were used. To develop the "Registration" module tools software as Prime Face Java, PostgreSQL, Enterprise Architect were so.

The following results for the indicators were obtained: Evaluation based on the type of requirements, UWE reached 66,67% and 100% OOWS; Assessment based on the functional proposal, UWE scored 33,33% while 66,67% OOWS achieved; Evaluation of design time, UWE reached 61,82% while 80,87% OOWS obtained; Evaluation of product model: UWE with 78,79% and 93,94% OOWS got; Assessment process model: UWE won 75,76% while 90,91% OOWS reached; Evaluation function points, UWE scored 66,77% and 73,10% OOWS. With descriptive statistics application the end result was achieved choosing the OOWS method that obtained the highest score in productivity.

GLOSARIO

Aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

Clase-Responsabilidad-Colaborador (CRC): identificación de verbos y sustantivos, entre otras técnicas, que permiten determinar las clases, métodos y atributos.

Contexto: Entidad que posee un conjunto de circunstancias (materiales o abstractas) que se producen alrededor de un hecho, o evento dado.

Controlador: Según el patrón MVC el controlador maneja la funcionalidad involucrada desde que el usuario entra a la web hasta que este sale, cada clic que ejecuta la conexión pasa por este componente, y según lo que suceda ejecutará la página JSP para mostrarle una salida al usuario.

Enterprise Architect: Herramienta de análisis y diseño con UML, que posee un modelado avanzado para negocios y sistemas de software.

Flujos de trabajo (Workflow): Estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

Glassfish: Servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación, es gratuito y de código libre, se distribuye bajo la licencia CDDL* y la GNU GPL*

Interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora.

Modelo de datos es un lenguaje orientado a hablar una Base de Datos.

Metodología hace referencia al conjunto de procedimientos racionales utilizados para alcanzar una gama de objetivos.

Modelo del producto: Es una abstracción de las propiedades comunes que se encuentran en cualquier producto desarrollado con el uso de un método, permite capturar los conceptos y patrones arquitecturales genéricos de cualquier aplicación web.

Navegación: Facilidad con la que un usuario puede desplazarse por todas las páginas que componen una aplicación web.

Programación es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales.

Puntos de control: Son mecanismos que verifican que se cumplen las condiciones o requerimientos necesarios para permitir el paso de una fase a otra.

Productividad: Definida como la relación entre los resultados y el tiempo utilizado para obtenerlos: cuanto menor sea el tiempo que lleve obtener el resultado deseado, más productivo es el sistema.

Primitivas de Abstracción: Se define como primitiva a los principales elementos que permiten el diseño de una aplicación web.

Requerimientos: es una necesidad documentada sobre el contenido, forma o funcionalidad de un sistema o un software.

ANEXOS

ANEXO 1

Calculo del tiempo promedio de demora en diseñar diagramas en la herramienta Case EA.

EVALUACIÓN DEL TIEMPO DE DISEÑO

Promedio de tiempo del diseño de cada diagrama

Tiempo del diagrama de clases: **1 hora**

Tiempo del diagrama de casos de uso: **0,6 hora**

Tiempo del diagrama de navegación: **1 hora**

Tiempo del diagrama de secuencia: **0,7 hora**

Tiempo del diagrama de presentación: **0,8 hora**

Promedio Tiempo: $1 + 0,6 + 1 + 0,7 + 0,8 = 4,1 / 5 = \mathbf{0,82 \text{ hora}}$

ANEXO 2

Tablas de valoración para el cálculo de estimación de esfuerzo que serán representadas en la herramienta COCOMO.

ESTIMACION DE ESFUERZO POR PUNTOS DE FUNCION

ILF: INTERNAL LOGIC FILE

Fichero lógico interno	N° DET	N° RET	Complejidad
MENU	4	6	MEDIA
MATRICULA	22	4	MEDIA
PADRES	30	6	ALTA
REPRESENTANTE	11	2	BAJA

EIF: EXTERNAL INTERFACE FILE

Fichero lógico externo	N° DET	N° RET	Complejidad
MENU	4	6	MEDIA
MATRICULA	22	8	ALTA
PADRES	30	10	ALTA
REPRESENTANTE	11	6	ALTA

EI: EXTERNAL INPUT

Entrada externa	Función	Número de entradas
El sistema permite ingresar los datos de los alumnos.	Alta/Modificación	13
El sistema permitirá ingresar los datos de los padres de familia.	Alta/Modificación	28
El sistema permitirá ingresar los datos del representante.	Alta/Modificación	10
El sistema permitirá ingresar los datos de la matrícula.	Alta/Modificación	3

Entrada externa	N° FTR	N° DET	Complejidad
El sistema permite ingresar los datos de los alumnos.	1	13	Baja
El sistema permitirá ingresar los datos de los padres de familia.	2	28	Alta
El sistema permitirá ingresar los datos del representante.	2	10	Alta
El sistema permitirá ingresar los datos de la matrícula.	1	3	Baja

EO: EXTERNAL OUTPUT

Salida externa	Función	Número de entradas
Permitirá mostrar mensajes de confirmación de guardado.	Pantalla	8
Permitirá mostrar mensajes de error o advertencia	Pantalla	13
Permitirá obtener un resultado o reporte de los alumnos matriculados.	Pantalla/Papel	2

Salida externa	N° FTR	N° DET	Complejidad
Permitirá mostrar mensajes de confirmación de guardado.	3	8	Alta
Permitirá mostrar mensajes de error o advertencia	6	13	Alta
Permitirá obtener un resultado o reporte de los alumnos matriculados.	2	2	Media

EQ: EXTERNAL QUERY

Consulta externa	Función	Número de entradas
Permite consultar una inscripción previa existente para realizar la matrícula	Pantalla	1
Busca el curso y paralelo	Pantalla	2
Permite buscar si el padre existe	Pantalla	2
Permite consultar el tipo de representante	Pantalla	3

Consulta externa	ENTRADA		SALIDA	
	DET	FTR	DET	RET
Permite consultar una inscripción previa existente para realizar la matricula	1	1	20	2
Busca el curso y paralelo	2	1	8	2
Busca la existencia del padre	2	2	28	2
Busca la existencia del representante	3	1	7	2

Consulta externa	C. Entrada	C. Salida	Complejidad
Permite consultar una inscripción previa existente para realizar la matricula	Baja	Alta	Alta
Busca el curso y paralelo	Baja	Media	Media
Busca la existencia del padre	Media	Alta	Alta
Busca la existencia del representante	Baja	Media	Media

RESUMEN

Parámetro	Complejidad	Nº	Peso	Total
ILF	ALTA	1	15	15
	MEDIO	2	10	20
	BAJO	1	7	7
EIF	ALTA	3	10	30
	MEDIO	1	7	7
	BAJO		5	
EI	ALTA	2	6	12
	MEDIO		4	
	BAJO	2	3	6
EO	ALTA	2	7	14
	MEDIO	1	5	5
	BAJO		4	
EQ	ALTA	2	6	12
	MEDIO	2	4	8
	BAJO		3	
TOTAL puntos de función				136

ANEXO 3

Calculo del tiempo promedio de demora en desarrollar las plantillas de presentación de interfaz por cada metodología.

Calculo de líneas de código por hora de cada metodología.

TEMPLATE	LINEAS DE CODIGO	TIEMPO EN HORAS
Clase	80	0,3
Función	165	0,8
Controlador	310	1,5
Vista	393	3
TOTAL	948	5,6
PROMEDIO	237	1,4

Metodología UWE

TEMPLATE	LINEAS DE CODIGO	TIEMPO EN HORAS
Clase	90	0,2
Función	178	0,4
Controlador	320	1,2
Vista	403	2,2
TOTAL	991	4
PROMEDIO	247,75	1

Metodología OOWS

ANEXO 4

Calculo del tiempo total en horas demoradas en desarrollar los prototipo1 en líneas de código.

PROTOTIPO UWE

TEMPLATE	N°	LINEAS DE CODIGO PCT	TOTAL	TIEMPO TRANSCURRIDO
Clase	2	60	120	780
	1	165	165	
	1	115	115	
	4	95	380	6 HORAS
Función	2	115	230	844
	3	53	159	
	1	67	67	
	1	128	128	
	1	260	260	7 HORAS
Controlador	2	117	234	1252
	1	78	78	
	1	103	103	
	1	330	330	
	1	507	507	14 HORAS
Vista	1	150	150	1045
	1	26	26	
	3	73	219	
	1	650	650	20 HORAS
TOTAL	28		3921	47 HORAS

ANEXO 5

Calculo del tiempo total en horas demoradas en desarrollar los prototipo2 en líneas de código.

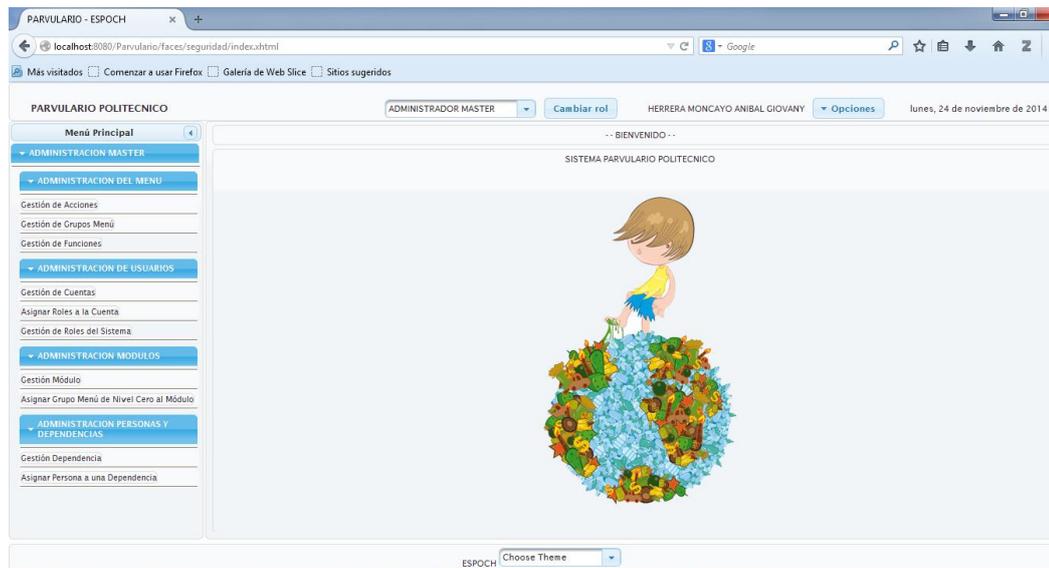
PROTOTIPO OOWS

TEMPLATE	N°	LINEAS DE CODIGO PCT	TOTAL	TIEMPO TRANSCURRIDO
Clase	1	143	143	751
	3	38	114	
	4	96	384	
	1	110	110	5 HORAS
Función	4	55	220	838
	2	115	230	
	1	128	128	
	1	260	260	7 HORAS
Controlador	1	82	82	1042
	2	110	220	
	1	465	465	
	1	275	275	13 HORAS
Vista	1	26	26	1050
	3	73	219	
	1	149	149	
	1	656	656	18 HORAS
TOTAL	28		3681	42 HORAS

ANEXO 6

Diagrama de Bases de Datos del Esquema Master_Movilización

La siguiente imagen muestra el modulo utilizado para el perfil de usuario Administrador Master, utilizado como ayuda para la organización del menú principal, acciones y funciones de usuarios del sistema parvulario.



BIBLIOGRAFIA

- Barrios J. (2002). «A component-based method for developing web applications.»
- Barrios, J., & Mendoza, M. (2004). Propuestas metodológicas para el desarrollo de aplicaciones Web: una evaluación según la ingeniería de métodos., *Vol. 25 N°2*, 90 - 96.
- Concepción, L. P. (2010). «Fundamentos de ingeniería de la web: ontologías, web semántica y agentes de software». *RISI*, vol. 7(n.o 1), pp. 77-89,.
- Del Valle Rodríguez, A. (2009). *Metodologías de Diseño usadas en Ingeniería Web, su vinculación con la NTics*. (Postgrado). Universidad Nacional de la Plata.
- Duque Escobar. (2011). « Metodologías de Desarrollo de Aplicaciones Web».
- Escalona, M., & Koch, N. (2002). «*Ingeniería de Requisitos en Aplicaciones para la Web—Un estudio comparativo*». Sevilla.
- Franco, Y. (s. f.). Tesis de Investigación : ESCALA DE LIKERT.
- Garmus, H. (2001). *Function Point Analysis. Measurement Practices for Successful Software Projects*. Ed. Addison Wesley.
- Kappel, G., Pröll, B., Reich, S., & Retschitzegger, W. (2006). *Web Engineering: The Discipline of Systematic Development of Web Applications*. John Wiley & Sons.
- Kraus, A., & Koch, N. (2002). «A metamodel for UWE». München.
- Loucopoulos, P., & Karakostas, V. (1995). *System Requirements Engineering*. McGraw-Hill.
- Mich, L., Franch, Mariangela, & Novi, I. (2004). «Market research for requirements analysis using linguistic tools». *Requir. Eng.*, vol. 9(n.o 1), pp. 40-56.
- Moreno L., L. (2011). «*AWA, marco metodológico específico en el dominio de la accesibilidad para el desarrollo de aplicaciones web*». Universidad Carlos III de Madrid.
- Niño, E. (2010). *Fundamentos para el desarrollo de aplicaciones en la red: Frameworks de desarrollo sobre páginas JSP*. Univesidad del Norte.

- Ocampo, E., & Ramón, G. (2009, abril). «*Desarrollo del sistema de control de proyectos para Santos CMI CONSTRUCTION INC utilizando la metodología OOWS*».
- Oliveros, A., Wehbe, R., Del Valle Rojo, S., & Rousselot, J. (2011). «Requerimientos para aplicaciones web».
- Pastor, A., & Fons Joan. (2002). Object-Oriented Web Solution (OOWS).
- Pérez, M., Méndez, E., Grimán, A., & Mendoza. (s. f.). ¿Cómo se relacionan la Calidad Sistémica y la Productividad en el Proceso de Desarrollo de Software? *Dpto. de Procesos y Sistemas, LISI. Universidad Simón Bolívar*.
- Presedo, C., Aguirregoitia, A., & Doblado, J. (2010). «Estudio de métricas para el control de proyectos software». *Actas 10º Taller Las Jorn. Sobre Apoyo Decisión En Ing. Softw. Bases Datos, vol. 4*.
- Pressman R. S., & Troya J. M. (1988). *Ingeniería del software* (McGraw Hill).
- Quintero, R. (2008). *Desarrollo dirigido por modelos de aplicaciones web que integran datos y funcionalidad a partir se servicios web*. Sevilla - Valencia.
- Rossainz, M., & Ocampo, J. (2001). Introducción a la Ingeniería Web Basada en UML. *Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación. Mexico*.
- Sommerville I. (2002). *Ingeniería del Software* (Sexta Edición.). Mexico: Editorial Addison Wesley.
- Torralba , J., Cuenca, J., & Ruíz , V. (2003). Métricas de Productividad para Gestión de Software. Universidad Politécnica de Valencia.
- Visconti Z, M. (2006). «Ingeniería de Software Avanzado».