



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO
PARA LA GESTIÓN DE LA INFORMACIÓN DEL CONTROL DE ACCESO A
LA SALA DE INTERNET DEL CTE DE LA UNACH.**

Trabajo de titulación presentado para optar al grado académico de:

INGENIERO EN SISTEMAS INFORMÁTICOS

AUTOR: LENIN CRISTIAN VELASTEGUÍ ORTIZ

TUTOR: DR. JULIO SANTILLÁN CASTILLO.

Riobamba-Ecuador

2016

©2016, Lenin Cristian Velasteguí Ortiz

Yo Lenin Velstegui Ortiz con cedula No. 060426295-6, certifico que las ideas expuestas y contenido en el presente trabajo de titulación son propias, y autorizo el uso del mismo ya sea para fines de investigación y/o académicos, siempre y cuando se registre el derecho de autoría.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal del trabajo de titulación certifica que: El proyecto técnico: “DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LA INFORMACIÓN DEL CONTROL DE ACCESO A LA SALA DE INTERNET DEL CTE DE LA UNACH”, de responsabilidad del señor Lenin Cristian Velasteguí Ortiz, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Luna DECANO DE LA FACULTAD DE INFORMATICA Y ELECTRONICA	_____	_____
Ing. Patricio Moreno. DIRECTOR DE LA ESCUELA DE INGENIERIA EN SISTEMAS	_____	_____
Dr. Julio Santillán Castillo. DIRECTOR DEL TRABAJO DE TITULACION	_____	_____
Ing. Raúl Rosero MIEMBRO DEL TRIBUNAL	_____	_____

“Yo, LENIN CRISTIAN VELASTEGUI ORTIZ, soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de titulación y el patrimonio intelectual del mismo pertenecen a la Escuela Superior Politécnica De Chimborazo”

Lenin Cristian Velastegui Ortiz

DEDICATORIA

Este logro lo dedico a toda mi familia en especial a mis padres, por todo el apoyo incondicional y cariño que me ha brindado durante toda mi vida, a mi tía Elvita por estar siempre en las buenas y en las malas y como no a mis grandes amigos por los consejos y por siempre apoyarme en mis decisiones.

Lenin Velastegui.

AGRADECIMIENTO

A la Escuela Superior Politécnica de Chimborazo por abrirme sus puertas y guiarme a través de sus profesionales para culminar mi carrera profesional y por medio de los conocimientos adquiridos aportar para progreso país. A mi familia y amigos por siempre apoyarme durante mi formación académica y al Centro de Tecnologías Educativas de la UNACH por darme la oportunidad de realizar mi trabajo de titulación.

Lenin Velastegui.

TABLA DE CONTENIDO

CONTENIDO	PAGINAS
PORTADA	
CERTIFICACIÓN DE AUTOR.....	I
CERTIFICACIÓN.....	II
DECLARACIÓN DE RESPONSABILIDAD	III
DEDICATORIA.....	IV
AGRADECIMIENTO	V
TABLA DE CONTENIDOS.....	VI
INDICE DE ABREVIATURAS	IX
INDICE DE TABLAS.....	X
INDICE DE ILUSTRACIONES.....	XII
RESUMEN	XIV
SUMMARY	XV
INTRODUCCIÓN	1
CAPITULO I	
1. MARCO TEORICO REFERENCIAL	5
1.1 SISTEMAS DE CONTROL DE ACCESO	5
1.1.1 Definiciones.....	5
1.1.2 Principios Básicos de un sistema de control de acceso.	6
1.1.2.1 Identificación.....	6
1.1.2.2 Autenticación.....	7
1.1.2.3 Autorización.....	7
1.1.3 Objetivos de un sistema de control de acceso	7
1.1.4 Tipos de Sistemas de control de acceso	8
1.1.4.1 Sistemas de control de acceso automática.....	8
1.1.4.2 Sistemas de control de acceso por red.....	8
1.2 KIT DE DESARROLLO DEL SISTEMA	9
1.2.1 Visual Studio	9
1.2.1.1 Definición	9

1.2.1.2	Características de Visual Studio	10
1.2.2	Node.Js	12
1.2.2.1	Definición	13
1.2.2.2	Motor V8	14
1.2.2.3	Tareas Asincronas.....	15
1.2.2.4	Npm.....	16
1.2.3	Socket.Io	18
1.2.4	MONGODB	19
1.2.4.1	Que es Mongodb.....	19
1.2.4.2	Documentos de tipo JSON, BSON	22
1.2.4.3	Almacenamiento orientado a documentos	22
1.2.4.4	Índices en Mongodb	22
1.2.5	Librería SocketIoClientDoNet	24
1.3.	INTERFAZ DE USUARIO PARA PANTALLAS TÁCTILES	24
1.4.	METODOLOGÍA SCRUM.....	24
1.4.1.	Características de SCRUM.....	25
1.4.2.	Roles de SCRUM	26
1.4.3.	Ciclo de vida de SCRUM.....	27
1.4.4.	SCRUM vs metodologías tradicionales.....	28
1.4.5.	SCRUM vs metodologías ágiles.....	29
1.4.6.	Ventajas de SCRUM	30
1.4.7.	Limitaciones deSCRUM	31

CAPITULO II

2.	MARCO METODOLOGICO	33
2.1.	FASE DE PLANIFICACIÓN	33
2.1.1.	Reuniones.....	35
2.1.2.	Procesos a automatizar	37
2.1.3.	Personas y roles del proyecto	38
2.1.4.	Pila del producto.....	38
2.1.5.	Análisis económico	40
2.2.	FASE DE DISEÑO.....	41
2.2.1.	Diagrama de Procesos	41
2.2.2.	Diagrama de casos de uso.....	42
2.2.3	Diagrama de clases.....	44
2.2.4.	Arquitectura de la aplicación.....	45
2.2.5.	Recursos necesarios.....	47
2.2.6.	Estándar de codificación	48
2.2.7.	Diseño de la interfaz de usuario	48
2.2.8.	Diseño de la base de datos.....	55
2.3.	FASE DESARROLLO E IMPLEMENTACIÓN.....	56
2.3.1.	Pila del sprint	56
2.3.2.	Tablero de tareas	59

2.3.3.	Pruebas de aceptación	60
2.4.	IMPLANTACION	61

CAPITULO III

3.	MARCO DE RESULTADOS	62
3.1.	GESTIÓN DEL PROYECTO	62
3.2.	APLICACIÓN DEL ESTANDAR ISO 9126.....	70
3.3.	VALIDACIÓN USABILIDAD	70
	CONCLUSIONES.....	74
	RECOMENDACIONES.....	75
	BIBLIOGRAFÍA.....	76
	ANEXOS.....	81
	ANEXO A: SPRINT DEL PROYECTO	81
	ANEXO B: DESARROLLO DEL SPRINT	82
	ANEXO C: DICCIONARIO DE DATOS.....	98
	ANEXO D: PRUEBAS DE ACEPTACIÓN	99

INDICE DE ABREVIATURAS

UNACH	Universidad Nacional de Chimborazo.
CTE	Centro de tecnologías estudiantiles.
NPM	Node Package Manager.
JS	JavaScript
SW	Software
HW	Hardware
BD	Base de Datos.
IO	Entrada/Salida
ESPCOH	Escuela Superior Politécnica de Chimborazo
GUI	Graphical user interface (en español Interfaz Gráfica de Usuario).
C/S	Cliente-Servidor
Pc	Computador personal
FIE	Facultad de Informática y Electrónica
IDE	Integrated development environment (Entorno de desarrollo integrado).
DBMS	Database Management System (Sistema de administración de base de datos).
SYSCONT	Sistema de Control
API	Application Program Interface (interfaz de programación de aplicaciones).

INDICE DE TABLAS

Contenido	Páginas
Tabla 1 – 2: Reuniones fase de planificación	35
Tabla 2 – 2: Roles del sistema.....	38
Tabla 3 – 2: Pila del producto	38
Tabla 4 – 2: Presupuesto del proyecto.....	40
Tabla 5 – 2: Recursos necesarios	47
Tabla 6 – 2: Estándar de codificación	48
Tabla 7 – 2: Pila del sprint	56
Tabla 8 – 2: Sprint 1 – Acciones del administrador	57
Tabla 9 – 2: Historia de Usuario HU01-S1	58
Tabla 10 – 2: Tablero de tareas	59
Tabla 11 – 2: Prueba de ingreso al sistema administrador.	60
Tabla 1 – 3: Horas de trabajo por semana	62
Tabla 2 – 3: Tiempos reales y estimados por semana	63
Tabla 3 – 3: Funcionalidad	65
Tabla 4 – 3: Confiabilidad	66
Tabla 5 – 3: Utilidad	66
Tabla 6 – 3: Eficiencia	67
Tabla 7 – 3: Mantenibilidad	68
Tabla 8 – 3: Portabilidad	69
Tabla 9 – 3: Identidad	71

Tabla 10 – 3:	Contenido	71
Tabla 11 – 3:	Utilidad	72
Tabla 12 – 3:	Navegación	72
2Tabla 14 – 3:	Retroalimentación	72

INDICE DE ILUSTRACIONES

Contenido	Páginas
Figura 1-1: Sistema de control de acceso	6
Figura 2-1: Motor V8	14
Figura 3-1: Evolución Node.js	15
Figura 4-1: Varias Peticiones a Node.Js.....	15
Figura 5-1: Tarea Asíncrona	16
Figura 6-1: Rpm instalados en servidor Node.js	17
Figura 7-1: Diferencias de bases SQL y NoSQL.....	21
Figura 8-1: Ejemplo entidad – relación.	21
Figura 9-1: Base de datos no relacionales documento JSON.....	21
Figura 10-1: Consulta en MongoDB.....	23
Figura 11-1: Procesos SCRUM	26
Figura 1-2: Actividades Planificadas	34
Figura 2-2: Diagrama Gantt de actividades	34
Figura 3-2: Diagrama de Procesos.....	42
Figura 4-2: Diagrama de casos de uso administrador.	43
Figura 5-2: Diagrama de casos de uso cliente.	43
Figura 6-2: Diagrama de clases administrador	44
Figura 7-2: Diagrama de clases cliente	45
Figura 8-2: Arquitectura del sistema.	46
Figura 9-2: Arquitectura aplicada al sistema de control de accesos	47
Figura 10-2: Interfaz del sistema – Pantalla de login	49
Figura 11-2: Pantalla principal.	50
Figura 12-2: Pantalla de login.	50
Figura 13-2: Panel computadores disponibles.	51
Figura 14-2: Acciones del Administrador	52

Figura 15–2:	Pantalla pc aleatoria	53
Figura 16–2:	Pantalla login del lado cliente	53
Figura 17–2:	Reportes Usuarios por fechas	54
Figura 18–2:	Reporte usuarios bloqueados	55
Figura 19–2:	Diseño de Base de datos	55
Figura 1–3:	Burndown Chart	63
Figura 2–3:	Indicadores del estándar ISO 9126	64
Figura 3–3:	Utilización de recursos por parte del sistema	67
Figura 4–3:	Análisis estándar ISO 9126	70
Figura 5–3:	Análisis de usabilidad	73

RESUMEN

El presente proyecto técnico contiene el desarrollo e implementación de un sistema informático para la gestión de la información del control de acceso a la sala de internet del Centro de Tecnologías Educativas para la Universidad Nacional de Chimborazo (UNACH) que se encuentra ubicada en el cantón de Riobamba, Provincia de Chimborazo, y que cuenta con alrededor de 3850 alumnos los cuales hacen uso de los laboratorios que dispone la institución. Por medio de Node.js tecnología la cual permite emplear la arquitectura cliente/servidor además permite aplicar javascript del lado del cliente como también del lado del servidor. Para que la aplicación administradora controle los computadores (clientes) se utilizó la librería socket.io, misma que permite la interacción bidireccional en tiempo real entre cliente y servidor, por medio de Visual Studio con C Sharp como lenguaje de programación se pudo programar la aplicación administrador y cliente, para el almacenamiento de los datos se usó Mongo.db base de datos que permite guardar datos en forma de documentos sin la necesidad de aplicar la regla de entidad relación. La metodología para el desarrollo de software fue SCRUM, metodología que permitió un adecuado entendimiento entre el cliente y el equipo de desarrollo, puesto que permite una activa participación por parte del cliente dentro de todo el proceso de desarrollo del sistema, formando parte activa del mismo. Al final el sistema podrá satisfacer todos los requerimientos establecidos al inicio del proyecto, respetando los tiempos establecidos dentro de la planificación, lo que permitió que el sistema se utilice de manera satisfactoria en el laboratorio de internet del Centro de Tecnologías Educativas, además se recomienda analizar y guardar la documentación expuesta en el presente trabajo de titulación para que informe de los requerimientos técnicos del sistema.

Palabras Clave: <APLICACION DE ESCRITORIO> <ARQUITECTURA CLIENTE / SERVIDOR> <ENTORNO DE DESARROLLO INTEGRADO .NET > <TECNOLOGIA NODE.JS> <LIBRERÍA DE NODE.JS SOCKET.IO> <METODOLOGIA DE DESARROLLO DE SOFTWARE SCRUM> <UNIVERSIDAD NACIONAL DE CHIMBORAZO (UNACH)> <CENTRO DE TECNOLOGIAS EDUCATIVAS (CTE)> <SISTEMAS INFORMATICOS>

SUMMARY

This technical project contains the development and implementation of a computer system for the management of information on access control to the Internet room of the Center for Educational Technologies for the National University of Chimborazo (UNACH), located in the canton of Riobamba, Province of Chimborazo, and which has about 3850 students who make use of the laboratories available to the institution. By means of Node.js technology which allows to use the architecture client / server in addition allows to apply javascript of the side of the client as well as of the side of the server. For the administration application to control the computers (clients) was used the socket.io library, which allows real-time bidirectional interaction between client and server, through visual studio 2012 with C Sharp as programming language could be programmed Application manager and client, for the data storage was used mongo.db database which allows to save data in the form of documents without the need to apply the relationship entity rule. The methodology for software development was SCRUM, a methodology that allowed an adequate understanding between the client and the development team, since it allows an active participation by the client within the entire process of development of the system, forming an active part of it. In the end, the system will be able to satisfy all the requirements established at the beginning of the project, respecting the established times within the planning, which allowed the system to be used in a satisfactory way in the Internet laboratory of the Center of Educational Technologies, in addition it is recommended to analyse and keep the documentation set forth in the present titling work to report the technical requirements of the system.

Keywords: <DESKTOP APPLICATION> <CLIENT / SERVER ARCHITECTURE>
<INTEGRATED NETWORK DEVELOPMENT ENVIRONMENT> <NODE.JS TECHNOLOGY>
<NODE.JS SOCKET.IO BOOKSTORGE> <SCRUM SOFTWARE DEVELOPMENT
METHODOLOGY> <NATIONAL UNIVERSITY OF CHIMBORAZO (UNACH)> <CENTER OF
EDUCATIONAL TECHNOLOGIES (CTE)> <INFORMATICS SYSTEMS>

INTRODUCCIÓN

La constante demanda de educación de calidad a nivel del Ecuador ha venido produciendo un impacto significativo en la manera en la que una institución educativa actúa en función de cubrir dichas demandas, así como también proporcionar mejores servicios y comodidades a los estudiantes, mientras en el pasado hablábamos de realizar cualquier tipo de operación de forma manual y de forma presencial por parte de alguna o varias personas encargadas de dicha tarea, ahora en los tiempos modernos todo se lo está realizando de manera automatizada ya sea a través de sistemas informáticos o de robots, eliminando así la intervención de las personas y sus varias equivocaciones.

La Universidad Nacional de Chimborazo (UNACH), se encuentra ubicada en el cantón de Riobamba, Provincia de Chimborazo, cuenta con alrededor de 3850 alumnos y tiene como misión “La Universidad Nacional de Chimborazo es una institución de educación superior, que en el marco de una autonomía responsable y de rendición social de cuentas, forma profesional emprendedores, con bases científicas y axiológicas, que contribuyen en la solución de los problemas del país”.

Entre uno de los departamentos importantes se encuentra el Centro de Tecnologías Educativas “CTE” que es un organismo académico administrativo de apoyo para la formación profesional de los estudiantes, brinda servicios de capacitación, consulta, investigación, soporte, logística, mantenimiento de equipos y asesoramiento técnico e interconectividad al personal docente y administrativo de la institución en las áreas de Computación, Informática, Desarrollo de Sistemas Informáticos y Telecomunicaciones.

El CTE está conformado por infraestructura costosa, así como también de equipos que son de manejo delicado, material informático de alto valor, tanto en equipos como en información, que en caso de pérdida o daño ocasiona un grave problema para la institución.

El CTE dispone de un laboratorio designado para la utilización por parte de los estudiantes que acuden diariamente a realizar diferentes tareas educativas como son: consultas en internet, trabajos escritos, utilización de programas, descarga y subida de archivos entre otros.

La ausencia de un sistema que automatice la gestión de información del control de acceso de los estudiantes hacia la sala de internet del centro de tecnologías educativas (CTE) de la UNACH limita el monitoreo y control de que el uso de los computadores sean de uso exclusivo de los estudiantes de la universidad.

Tampoco al no tener un registro confiable de la información del uso de los computadores por parte de los estudiantes existen problemas al momento de la toma de decisiones por parte del personal administrativo del CTE para así mejorar sus servicios.

Además, uno de los principales y más grandes problemas es que debido al gran número de estudiantes que acuden a la sala de internet, es muy tedioso llevar este control de acceso de forma manual ya que conlleva mucho tiempo llenar los datos de los estudiantes que desean ingresar a la sala de internet.

Formándose en el peor de los casos colas de estudiantes y esto implica malestar por parte de las personas que necesitan del uso de los computadores que se encuentran en el laboratorio del CTE, debido a esto, surge la necesidad de desarrollar alternativas que permitan resolver el problema de control de acceso de los estudiantes a dichas instalaciones, haciendo uso de la tecnología actual, para así explotar al máximo sus capacidades.

El Objetivo principal de cualquier sistema informático de control de acceso es el de dotar los mecanismos de seguridad adecuados a las dependencias que se intenta proteger; este conjunto de mecanismos debe incluir al menos un sistema que permita identificar a las entidades (generalmente usuarios) que intentan acceder a los objetos (la empresa, institución, departamento, etc.), mediante algún proceso de identificación.

Atraves del desarrollo e implementación del sistema de control de accesos se busca la automatización de procesos que ayuda a optimizar recursos tangibles e intangibles, un recurso intangible y de gran valor es el tiempo, con la implementación de un sistema que gestione la información para el control de accesos se pretende agilizar los procesos realizados en la sala de internet, procesos tales como apagado, reinicio, bloqueo, desbloqueo de computadores, además permite llevar un registro confiable de toda la información de uso de los computadores en una base de datos, información que es importante para futuros reportes de uso.

Actualmente el uso de herramientas para la integración de procesos en el desarrollo de sistemas informáticos, han permitido construir software de calidad por ese motivo se realizó una revisión previa para elegir las más adecuadas.

Elementos Hardware

- ✓ Computador Hp core i7 All in one – Máquina Central.
- ✓ Computadores de escritorio Hp core i7 – Clientes.
- ✓ Computador Hp core i7 – Servidor.
- ✓ Portátil.
- ✓ Cable de red.

Elementos Software

- ✓ Visual Studio.- permite el uso de un amplio rango de tecnologías de desarrollo tanto para aplicaciones de escritorio, orientados a la web, móviles, entre otros.
- ✓ MongoDB. - Herramienta que presenta una nueva forma de utilizar base de datos de una manera no relacional, esto permite que el sistema sea adaptable y flexible.
- ✓ Node.js. - Permite construir aplicaciones rápidas, escalables y que soportan conexiones concurrentes prácticamente infinitas de la mano con su librería sockets.io para realizar una comunicación servidor/cliente de manera inmediata.
- ✓ JSON.Net.- Herramienta que permite la comunicación de diferentes sistemas independientemente del lenguaje de programación, se lo aplico para la comunicación entre C# y Node.js

En cuanto a la metodología de desarrollo de software se optado por la metodología ágil SCRUM puesto que estas permiten la intervención directa del usuario durante el desarrollo del software, ofrecen una rápida respuesta a cambios de requisitos gracias a su proceso iterativo, permitiendo el desarrollo, testeo y correcciones rápidas que facilita la entrega de productos de calidad a tiempo.

Para el control de acceso de estudiantes a los laboratorios se tomó en consideración utilizar una interfaz táctil, debido a que el CTE cuenta con computadoras All in one, evitando así el uso del mouse y teclado, permitiendo dar una mejor experiencia de uso de la aplicación a los estudiantes y así acoplado el sistema a las últimas tendencias que son las interfaces de usuario táctiles.

CAPITULO I

1. MARCO TEORICO REFERENCIAL

1.1 Sistemas de Control de Acceso

Un sistema de control de acceso es un mecanismo electrónico cuyo objetivo es proporcionar los permisos necesarios para que los usuarios puedan tener o no el acceso a una determinada área, edificios, instalaciones o también a recursos digitales como archivos, sitios webs, equipos, etc.

La validación de la información para el acceso se la puede realizar mediante el uso de dispositivos de entrada, por medio de la digitación de códigos por teclado, lectores de proximidad, códigos de barra o sistemas biométricos, el sistema de control de accesos identifica al usuario y permite o deniega el acceso a determinado recurso, después de haber validado la información ingresada con los de su base de datos.

1.1.1 Definiciones.

Para tener un mejor entendimiento sobre los que es un sistema de control de acceso se debe definir sus componentes principales que son los siguientes.

- **Sujetos.** - Que son las entidades que pretenden utilizar los recursos del sistema y como consecuencia cambian el estado del sistema, las entidades pueden ser (usuarios, grupos, clientes de un banco, estudiantes, trabajadores de una empresa, etc.)
- **Recursos.** – Son todas las entidades que el sistema intenta proteger, a los cuales solo lo podrán usar los sujetos que tengas los permisos necesarios. Estos recursos pueden ser, archivos, datos, dispositivos de hardware, equipos de computación, ingreso a determinada área, etc.
- **Permisos.** – Son las acciones permitidas que un sujeto puede ejecutar sobre los recursos.

- **Solicitud de acceso.** – Es la acción que realiza el sujeto al sistema para que este le permita utilizar cierto recurso.

1.1.2 Principios Básicos de un sistema de control de acceso.

El concepto de un sistema de control de acceso está basado en tres principios básicos que son: la **identificación, autenticación y autorización**, el sistema utiliza estos tres principios para controlar si los recursos están disponibles o no a los usuarios que pretenden usarlos.

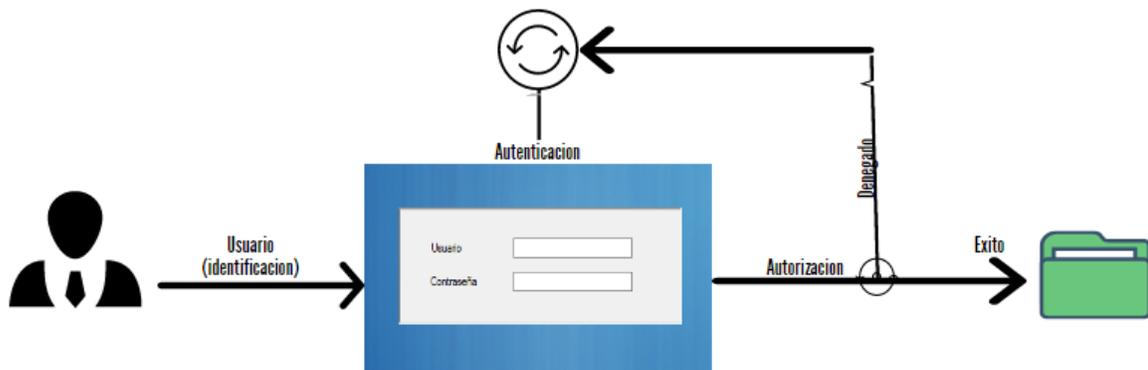


Figura 1-1. Sistema de control de acceso

Fuente: Velastegui , Lenin, 2016.

1.1.2.1 Identificación.

Este es el proceso por el cual permite al sistema identificar cual es el usuario que realiza una petición para ingresar al sistema. En términos generales este es el medio por el cual el usuario se da a conocer al sistema.

1.1.2.2 Autenticación

Este es el siguiente paso dentro del control de acceso, la autenticación es el proceso por el cual el sistema permite la validación de la identidad del usuario. Son comunes para este fin las contraseñas, lectores de huellas, reconocimiento facial, ocular, etc.

1.1.2.3 Autorización.

Una vez que el usuario se autentica de forma correcta el sistema le autoriza ocupar cualquier tipo de recurso que el sistema este custodiando, además que solo tendrá permisos de usar los recursos a los cuales tenga permitido utilizar de acuerdo al cargo que el usuario este desempeñando en la empresa o institución.

Por ejemplo, en una institución educativa dentro del sitio web para calificaciones, los estudiantes solo podrían ver sus notas, mientras que los profesores podrían ingresar, modificar y eliminar las notas de los estudiantes.

1.1.3 Objetivos de un sistema de control de acceso

Entre los varios objetivos por los cuales se implementan un sistema de control de acceso se exponen los siguientes:

- Denegar el acceso a la información (recursos) a usuarios que no han sido autorizadas. Caso contrario permitir el acceso solo a los recursos permitidos.
- Asegurar los recursos que se desean custodiar mediante la implementación de técnicas de autenticación y de autorización.
- Llevar un control de los eventos y actividades que realizan los usuarios dentro del sistema.
- Concientizar a los usuarios sobre la responsabilidad que tienen con el manejo de contraseñas y equipos.

- Garantizar la seguridad de los recursos en horarios en los cuales no se permitido utilizar.

1.1.4 Tipos de Sistemas de control de acceso

Los sistemas de control de acceso se dividen básicamente en dos tipos que son:

- Sistemas de control de acceso automática
- Sistemas de control de acceso en red.

1.1.4.1 Sistemas de control de acceso automática.

Estos sistemas de control son los más simples, permiten controlar lo que es una o más puertas, esto sin estar conectados a ningún sistema central que lo controle o conectado alguna computadora donde pueda guardar la información de los usuarios.

Por ende, no permite llevar un seguimiento detallado de las actividades que se realizan con el sistema, además algunos sistemas de control de acceso no permiten controlar el acceso por horarios o grupos de puertas, esto en si depende de las marcas y de la robustez de sus productos.

1.1.4.2 Sistemas de control de acceso por red.

Estos sistemas son más complejos que los anteriores debido a que estos van conectados a un sistema central, que puede ser una computadora local o también puede estar controlada por alguna computadora de forma remota.

La computadora central lleva un software que le permite llevar un control de las actividades realizadas por el usuario y guardarlas en una base de datos donde posteriormente se puede realizar consultas de dicha información.

Estas actividades se las guarde de forma detallada por ejemplo nombres de usuarios, cédula de identidad, horario de ingreso y de salida etc., estos sistemas van desde los más sencillos hasta mega sistemas muy complejos y sofisticados esto depende de lo que se requiera.

1.2 Kit de desarrollo del sistema

A continuación, se detallan todas las herramientas que se utilizaron para el desarrollo del presente proyecto.

1.2.1 Visual Studio

La utilización de Visual Studio es necesaria para la creación de software debido a su fácil manejo del diseño de formularios y la utilización de sus componentes, por la documentación, ayudas y videos explicativos que existen en la web, además de la extensa lista de librerías que existen y que han sido desarrolladas para este IDE, librerías como la de SocketIoClientDotNet la cual es necesaria para poder utilizar las funcionalidades de socket.io en cada uno de los computadores en los cuales se van a instalar el sistema, otra librería es la de mongodb la cual permite la interacción entre la aplicación desarrollada con la base de datos permitiendo acciones como guardar consultar y actualizar la información , estas librerías son de fácil instalación debido a que Visual Studio cuenta con su gestor de librerías el cual se lo conoce como Nuget este permite realizar búsquedas y descargar las librerías que se vaya utilizar directamente desde el internet o instalarlas de forma manual.

El lenguaje de programación elegido es C Sharp “C#” y Visual Studio es el mejor IDE que permite la utilización de este tipo de lenguaje de programación de forma sencilla ya que cuenta con un editor de código el cual permite diferenciar partes de código a través de colores para poder diferenciar métodos, clases, variables, etc.

Debido a que el sistema a desarrollar se lo va implementar en computadoras que van a contener sistema operativo Microsoft Windows en sus versiones 7, 8 y 10 no es necesario del uso de un IDE que permita programar sistemas para múltiples plataformas y en cuanto a los seriales de activación se refiere la UNACH cuenta con las licencias para la utilización de forma legal de visual estudio en su versión del 2012.

1.2.1.1 Definición

“Visual Studio es un IDE (en español entorno de desarrollo integrado) perteneciente a Microsoft, el cual solo se puede utilizar en dispositivos que corran sistema operativo Windows, soporta varios

lenguajes de programación como: C#, C++, f#, Visual Basic .Net, Java, Ruby, Python, PHP, además permite programar en entornos de desarrollo para web como ASP.NET, MVC, Django, etc.

Visual Studio es utilizado por los programadores de hoy en día para desarrollar aplicaciones de escritorio, para móviles, sitios y aplicaciones webs, así como también de servicios en la web. “ (1)

1.2.1.2 Características de Visual Studio

- ✓ El depurador (debugger), Este permite analizar línea por línea en búsqueda de errores en las aplicaciones programadas con cualquier tipo de código que soporte Visual Studio. Así también se pueden realizar la depuración a los procesos que se ejecuten en el computador. El debugger permite crear breakpoint o conocidos como puntos de ruptura los cuales permiten interrumpir de manera temporal la ejecución de la aplicación en una determinada posición permitiendo al programador realizar un análisis del actual estado de las variables en ese determinado punto de ruptura.
- ✓ El diseñador (Designer), como su nombre lo dice permite al programador diseñar la interfaz de usuario y observar la apariencia que esta tendrá antes de obtener el producto final. El designer puede adoptar varias formas entre las principales tenemos:
 - Windows forms, Esta clase de formularios se utiliza para el diseño de aplicaciones de tipo escritorio. Los controles como botones, labels, textbox, etc, se los arrastra sobre el formulario y se los enlaza directamente al origen de datos (código) o a los eventos de la aplicación.
 - WPF (Windows Presentation Foundation), este fue agregado desde la versión de Visual Studio 2008, y permite al igual que Windows forms colocar los controles directamente sobre los formularios que se estén implementando. Este diseñador para poder representar los controles sobre los formularios utiliza etiquetas XAML en lugar de lenguajes de programación por ejemplo C#.
 - Class designer, utilizando el modelamiento UML (Unified Modeling Language) este diseñador puede crear y editar clases, incluyendo sus miembros y su acceso. Esta herramienta es útil para generar diagramas de clases las cuales ya deben estar codificadas previamente.

- Data designer, Este diseñador permite crear bases de datos de forma gráfica y sus componentes como tablas, valores y relaciones entre tablas, además permite la realización de consultas.
- Mapping designer, Diseñador introducido con la versión de Visual Studio 2008, este permite al desarrollador crear y editar relaciones LINQ y también permite asociar que encapsulan la información.
- Web designer/development, con este diseñador podremos crear editar y diseñar todo tipo de páginas web por el método de Drag and Drop, lo que quiere decir que el programador puede arrastrar los componentes directamente sobre un formulario web y ubicar los componentes donde lo desee. Es utilizado para desarrollar aplicaciones ASP.NET y soporta lenguajes de programación tales como HTML, CSS y Java Script.
- ✓ IntelliSense, a través de esta se puede ingresar a las descripciones de cada una de las funciones y de sus correspondientes atributos. Permite llevar un desarrollo de software más eficiente debido a que intellisense reduce la cantidad de nombres que deben llevar en memoria y volumen de codificación requerida.
- ✓ Explorador de pestañas abiertas o en inglés (Open Tabs Browser), permite un desplazamiento por todas las pestañas en las que se esté trabajando en ese momento y así mantener una navegación del proyecto de una forma ordenada.
- ✓ Editor de propiedades, permite modificar las propiedades de los objetos representados en el diseñador (designer) de Visual Studio. Las todas las propiedades de los objetos tales como: clases, formularios, páginas web y otros elementos.
- ✓ Explorador de objetos, permite explorar nombre, objetos y todas las librerías de clases para Microsoft .Net. Puede ser utilizado para navegar los espacios de nombres en ensamblados administrados.
- ✓ Explorador de solución. Dentro de Visual Studio una solución está definida como un conjunto de archivos de código y otros recursos que son utilizados para construir una aplicación. Todos estos archivos son ordenados de forma jerárquica para facilitar la navegación mediante el explorador de soluciones.

- ✓ Explorador de servidores, nos permite gestionar las conexiones de bases de datos de un proyecto. También es utilizado para explorar eventos de log y servicios de Windows en ejecución, colas de mensajes, contadores de desempeño, y utilizarlos como un origen de datos.
- ✓ Visual Studio se caracteriza por permitir el desarrollo de aplicaciones de escritorio, sitios y aplicaciones web, también permite la creación de servicios web en cualquier entorno que se sea compatible con .NET, es decir que se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles, logrando cerrar un círculo que alcanza los avances tecnológicos tanto en hardware como en software.
- ✓ Microsoft ha evolucionado a lo largo del tiempo y siempre está pendiente de la nueva tendencia en tecnología es por eso que por medio de su IDE Visual Studio ha logrado desarrollar aplicativos para pantallas multitáctiles, tales como Windows Azure, Windows Phone 7 o Sharepoint.

1.2.2 NODE.JS

Node.JS no es una tecnología relativamente nueva (sus orígenes datan del año 2009) y no se le ha dado el lugar que se merece, tiene como objetivo principal el de proporcionar una manera fácil para desarrollar programas de red escalables.

Node.js es una tecnología la cual fue creada por Ryan Dahl, esta tecnología permite a los desarrolladores crear servidores de muy alto rendimiento gracias a que utiliza el motor JavaScript V8 de Google, con un modelo asíncrono de Entrada/Salida. Node.js permite desarrollar servidores de alta concurrencia, y se puede alcanzar un desarrollo web de alto rendimiento gracias a la gran lista de módulos los cuales han sido creados por terceros.

Node.js intenta resolver problemas en la utilización de recursos de memoria en comparación con otros lenguajes de programación, por ejemplo, si se desea implementar un servidor con Java o PHP, estos servidores consumen más memoria en comparación a un servidor creado con Node.js, debido a que cada nueva conexión que se realice a uno de los servidores de Java o PHP generan un nuevo proceso o un nuevo hilo.

Es decir, en el caso que dispongamos de un servidor (hardware) que tenga 8 Gb de memoria RAM, y se le ha configurado, que para cada conexión ocupe 2MB de memoria, esto quiere decir que solo podríamos tener un máximo de 4000 conexiones al servidor.

En caso que se necesite más conexiones, se debería añadir memoria RAM o en el peor de los casos agregar más servidores para poder así atender la demanda de usuarios. Esto genera varios problemas técnicos y estos se convierten en costos no planificados.

Entonces como Node.js da solución a este problema, lo resuelve modificando la manera en la que se realiza la conexión cliente/servidor, es decir que en vez de crear un hilo para cada conexión y asignar memoria, ahora cada conexión que se realiza se basa en la ejecución de eventos dentro de un solo proceso en el motor de V8 de Node.js, por lo que no existen puntos muertos debido a que no hay bloqueos al momento de realizar las conexiones, es por esto que Node.js garantiza que cualquier servidor que lo ejecute podrá tolerar miles de conexiones concurrentes.

1.2.2.1 Definición

“Node.js es una plataforma que se construyó con las rutinas de JavaScript para Chrome, con el objetivo de facilitar la construcción de aplicaciones rápidas y escalables en conexiones.

Utiliza un modelo I/O que no bloquea el sistema y está controlado por eventos. Gracias a este modelo, Node.js es tan ligero y tan eficiente, convirtiéndose en un lenguaje de programación ideal para programar aplicaciones que deban trabajar, en tiempo real, con una gran cantidad de datos ejecutándose en todo tipo de dispositivos.” (2)

A Node.js se la toma como una tecnología que agrupa varias herramientas las cuales son gestionadas por el motor V8 que utiliza JavaScript y que fue creado por google para su navegador Chrome, actualmente Node.js ha tenido una gran acogida por miles de programadores alrededor del mundo debido a que está diseñado para ejecutar código JavaScript de forma eficiente y extremadamente rápida, y lo mejor de todo es que lo permite la gestión de eventos desde el lado del servidor.

Node.js en si es tomada como un entorno de programación a través de JavaScript, y la única limitante que nos impone es la de nuestra imaginación y de nuestros conocimientos de programación.

Con Node.js podemos realizar desde pequeños scripts para que hagan en el sistema de archivos hasta la creación de aplicaciones webs robustas destinadas para empresas grandes, así como también para desarrollar apps de redes, sistemas que trabajen con peticiones que trabajen en tiempo real, chats de

todo tipo, juegos multijugadores, en si esta tecnología esta destina para trabajar en aplicaciones que contengas cientos de usuarios interactuando entre si y en tiempo real.

El manejo de las estructuras de manera más eficiente y diferente hace que el estudio de Node.js nos lleve al descubrimiento de nuevas tendencias en este caso para la mejora de interfaces gráficas, como para mejora en la parte de ejecución.

1.2.2.2 Motor V8

Google en su afán de ganar espacio en el mercado de los navegadores, creo el motor V8 para su navegador google Chrome, el V8 es un software muy optimizado el cual permite ejecutar código JavaScript de una forma extremadamente rápida, este fue diseñado especialmente para la web.

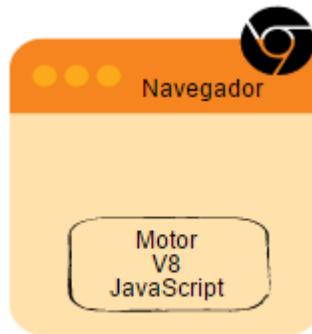


Figura 2-1. Motor V8

Fuente: Velastegui , Lenin, 2016.

Gracias a la tecnología con la que está compuesto Node.js, este nos permite utilizar el motor V8, pero ya no solo del lado del navegar sino ahora también desde el lado del servidor, abriéndonos así un mundo de posibilidades en cuanto al área del desarrollo de las aplicaciones se refieren.

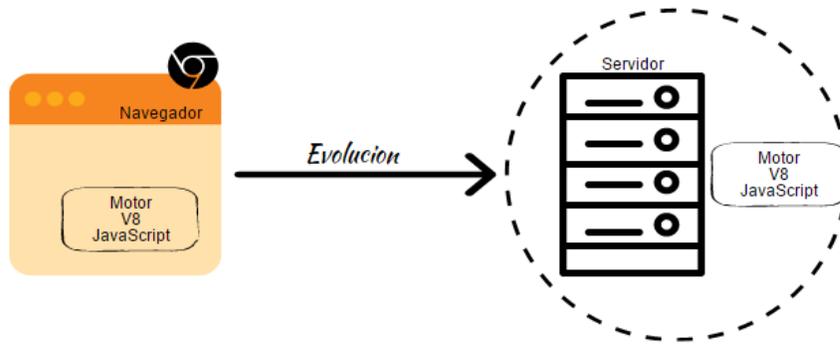


Figura 3-1: Evolución Node.js

Fuente: Velastegui , Lenin, 2016.

Para poder pasar a usarse JavaScript desde el lado del servidor, Node.js ha desarrollado una variedad de librerías propias la cuales no tienen que envidiar a ninguna otra plataforma de desarrollo de aplicaciones, así también ha permitido eliminar varias funciones del lado del servidor que no tienen nada de sentido usarlo como por ejemplo el **Document Object Model**.

1.2.2.3 Tareas Asíncronas

Node.js controla todo el flujo de un trabajo determinado a través de un solo hilo de ejecución, el cual está encargado de realizar dicha función, recibe todas las peticiones de los clientes y los pasa al hilo el cual realiza las tareas de forma asíncrona.

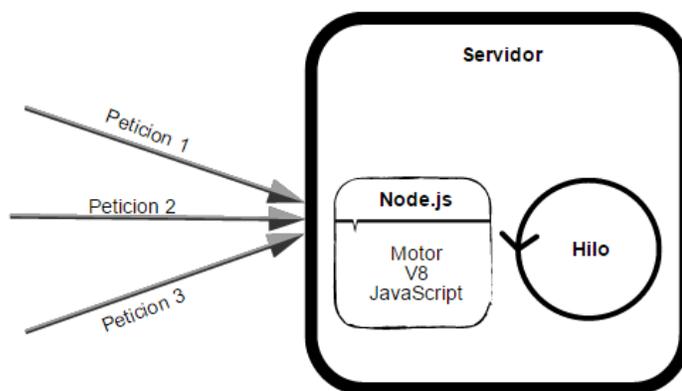


Figura 4-1: Varias Peticiones a Node.Js

Fuente: Velastegui , Lenin, 2016.

Node.js para procesar todas las peticiones que le llegan al mismo tiempo, las realiza de forma asíncrona (Quien envía continúa con su ejecución inmediatamente después de enviar el mensaje al receptor), para realizar este proceso Node.js utiliza recibe todas las peticiones y las procesa con su único hilo el cual delega las tareas a un pool de thread el cual permite crear hilos independientes para cada tarea, este pool de thread está construido con la librería libuv, Node.js envía el conjunto de peticiones al pool de threads.

Cuando alguna tarea dentro del pool thread se haya realizado la librería libuv envía un evento hacia Node.js, para notificar que la tarea se ha completado, este evento llega en forma de un callback, el cual permite que el proceso se termine, en si cuando trabajemos con Node.js prácticamente toda la programación que realicemos será de forma asíncrona ya que todo el trabajo lo delega al pool thread.

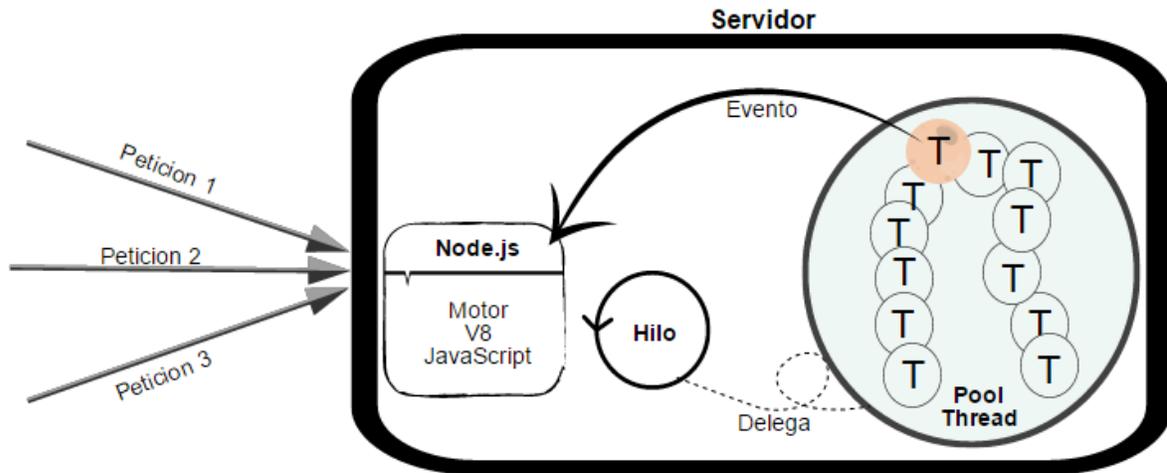


Figura 5-1: Tarea Asíncrona

Fuente: Velastegui, Lenin, 2016.

1.2.2.4 NPM

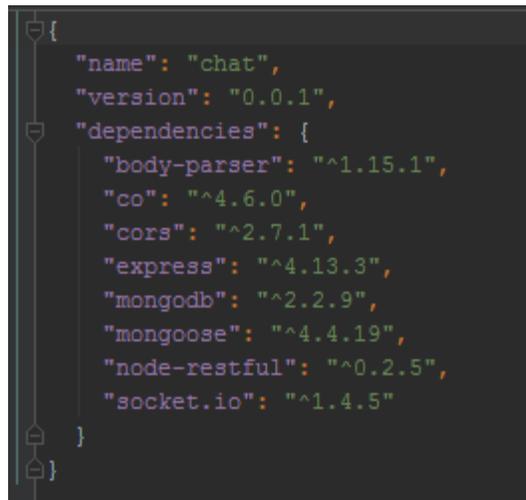
El npm es el gestor de paquetes para Node.js, conocido en inglés como Node Package Manager, por medio de este gestor los desarrolladores se puede descargar, crear y compartir módulos, estos módulos pueden ser reutilizados en las aplicaciones Node.js que se estén desarrollando, además también sirven para poder reutilizar aplicaciones Node.js totalmente completas, esto gracias a que la comunidad de Node.js publica la mayoría de sus módulos y aplicaciones con licencia open source, es decir podemos instalar, modificar y hasta distribuir libremente.

Los módulos en si son similares a las librerías de otros entornos de desarrollo, estos módulos se pueden descargar para cualquier aplicación que lo amerite, en comparacion con otros IDEs seria parecido a RubyGems en Ruby, CPAN en Perl, PIP en Phython o PEAR en PHP, existen módulos para realizar varias funciones como por ejemplo para, conexiones con bases de datos, para validar entradas de datos, para analizar archivos, entre otros.

El uso de los módulos de Node.js son de mucha utilidad para aquellas personas que están empezando o tienen poca experiencia en el desarrollo de aplicaciones, debido a que estos módulos permiten empezar a trabajar basándose en conocimientos de otros programadores. Independientemente de lo que deseemos desarrollar con Node.js siempre debemos conocer sobre la utilización de npm y con sus librerías que nos pone a nuestra disposición.

Para instalar módulos desde npm se debe primero instalar Node.js y después desde el cmd símbolo del sistema se debe ingresar el siguiente comando.

```
npm install [nombre_del_modulo]
```



```
{
  "name": "chat",
  "version": "0.0.1",
  "dependencies": {
    "body-parser": "^1.15.1",
    "co": "^4.6.0",
    "cors": "^2.7.1",
    "express": "^4.13.3",
    "mongodb": "^2.2.9",
    "mongoose": "^4.4.19",
    "node-restful": "^0.2.5",
    "socket.io": "^1.4.5"
  }
}
```

Figura 6-1: Rpm instalados en servidor Node.js

Fuente: Velastegui , Lenin, 2016.

Este comando manda una petición al servidor de npm para que nos permita descargar la última versión del módulo que necesitamos.

1.2.3 SOCKET.IO

Socket.io es una librería de Node.js codificada en su totalidad con JavaScript, esta librería permite la comunicación en tiempo real entre cliente/servidor. La comunicación la realiza a través de Websocket (utiliza el protocolo http para la comunicación y algún puerto disponible) pero también lo puede hacer con otro tipo de sockets tales como Adobe Flash, long polling en AJAX o JSONP polling, esto depende de los requerimientos del cliente en tiempo de ejecución. Para su instalación simplemente ejecutamos un comando de npm para poder instalar dentro de Node.js.

Una de las desventajas de las aplicaciones realizadas con socket.io es que no soportan comunicaciones con otros clientes los cuales estén usando los Websockets estandar. Esto se da debido a que Socket.io no es un protocolo sino es una librería de comunicación web en tiempo real que utiliza varios protocolos. Es útil cuando se desarrollan proyectos en los cuales se instala socket.io en los dos extremos de la comunicación es decir en el servidor y en el cliente.

Socket.IO es un módulo para Node.js que proporciona una forma sencilla de establecer una comunicación a través de WebSockets. El protocolo WebSocket es complicado, y escribir una aplicación desde cero que sea compatible con él nos llevara mucho tiempo, pero con la utilización de socket.io se simplifica la creación de aplicaciones compatibles con websockets.

Socket.IO proporciona componentes tanto para el servidor como para el cliente. Así pues, solo necesitaremos un módulo para que nuestra aplicación sea compatible con el protocolo WebSocket. Socket.IO también resuelve el problema de compatibilidad con los clientes y permite establecer comunicaciones en tiempo real.

El diseño de Socket.IO es muy bueno y simplifica muchísimo el proceso que permite a nuestra aplicación establecer comunicaciones en tiempo real. Si queremos hacer algo que necesite una comunicación en tiempo real entre un servidor y navegador, la mejor solución es combinar Node.js y Socket.IO.” (2).

Socket.IO es un módulo construido totalmente con JavaScript, permite programar tanto del lado servidor como del cliente, se trabaja así porque la comunicación es realizada en los dos sentidos, por esa razón Socket.IO debe ser programado en los dos extremos de la comunicación establecida.

1.2.4 MONGODB

Cuando se habla de NoSQL, no es solo de referirse a un tipo de base de datos, sino a varios tipos o métodos para almacenar datos, dichos métodos no cumplen con el esquema típico entidad-relación, y no imponen una estructura de datos fijas en forma de tablas o relaciones. Estos tipos de base de datos permiten almacenar los datos en diferentes formas como: llave-valor, documentos, xml, grafos, columnas entre otros.

Mongodb al ser parte de las bases de datos NoSQL, es un sistema de almacenamientos de datos que está pensado para el manejo de grandes cantidades de información de una manera muy ágil, escalando de manera horizontal, aumentando el rendimiento cuando se necesita crecer, y no como las bases de datos relacionales que escalan o crecen de manera vertical, es decir, añadiendo memorias o discos duros con mayor velocidad cuando se necesita crecer en almacenamiento de datos.

Los datos almacenados en Mongodb no necesitan una estructura fija (tablas), ahora su estructura puede variar en cualquier momento ya que se utiliza colecciones (equivalentes a las tablas en SQL) y sus documentos (equivalentes a los registros) que pueden variar dinámicamente en función a las necesidades, esto se convierte en una gran ventaja en comparación a las bases de datos relacionales SQL debido a que su estructura es fija y en la mayoría de los casos es complejo modificar la estructura ya que esto conlleva modificar todas las tablas que estén relacionadas al cambio realizado.

1.2.4.1. Que es Mongodb

MongoDB hace referencia al término «humongous» que en español significa gigantesco, esta herramienta de software es una de las principales bases de datos dentro de NoSQL, permite a las empresas ser más ágiles y escalables en cuanto al almacenamiento de datos se refiere. Todo tipo de organizaciones grandes, medianas y pequeñas están empezando a usar MongoDB para el desarrollo de nuevas aplicaciones, reducción de tiempo en el comercio, mejora de la experiencia de los usuarios y en si en la reducción de costos de la empresa.

A través de este tipo de NoSQL se puede modificar fácilmente los esquemas de las bases de datos de aplicaciones las cuales se encuentran en pleno desarrollo o pos-desarrollo, además, proveen a los desarrolladores las funcionalidades que cualquier base de datos tradicional pueden ofrecerles tales como: búsquedas, índices y consistencia estricta.

El objetivo de MongoDB es brindar escalabilidad, rendimiento y gran disponibilidad, proporcionando un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria. MongoDB brinda flexibilidad operativa y fiabilidad empresarial gracias a la replicación nativa y a la tolerancia a fallos.

MongoDB como base de datos no relacional permite que los sistemas pueden manejar grandes cantidades de datos sin ataduras como lo hacen las convencionales bases de datos relacionales que mantienen las reglas conocidas como entidad relación, ayuda en la flexibilidad de los sistemas permitiendo que se puedan utilizar en todo tipo de dispositivos.

Para diferenciar cada uno de los términos dentro de MongoDB se toma la siguiente descripción:

- Un sistema Mongo contiene un conjunto de base de datos.
- Una base de datos contiene un conjunto de colecciones.
- Una colección contiene un conjunto de documentos.
- Un documento es un conjunto de campos.
- Un campo es un par de clave/valor.
- Una clave es un nombre (cadena).
- Un valor es:
 - Un tipo cadena, entero, binario, etc.
 - Un documento o un vector de valores.

A continuación, se representa la diferencia de manejo de los datos entre base de datos relacionales y no relacionales.

Base de datos SQL	Base de Datos NoSQL
Tablas	Colecciones
Filas	Documentos
Índices	Índices

Figura 7-1: Diferencias de bases SQL y NoSQL.

Fuente: Velastegui , Lenin, 2016.

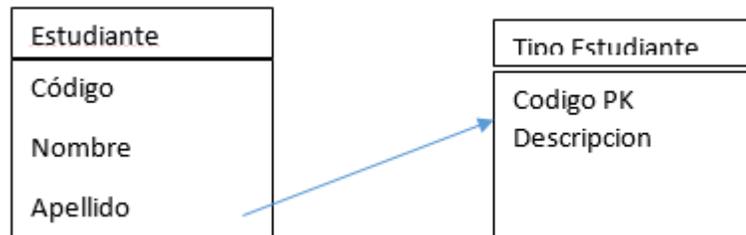


Figura 8-1: Ejemplo entidad – relación.

Fuente: Velastegui , Lenin, 2016.

```

//Colección Estudiante
{
  Codigo:"cod1",
  Nombre:"Nombre",
  Apellido: "Apellido",
  Sexo:"Masculino",
  Tipo_Estudiante:{
    Codigo:"cod1",
    Descripción:" Postgrado"
  }
}
  
```

Figura 9-1: Base de datos no relacionales documento JSON.

Fuente: Velastegui , Lenin, 2016.

En la figura 9-1 se observan los datos en forma de documentos JSON, los cuales siempre estarán escritos entre llaves y para diferenciar entre el nombre del campo y el valor que le corresponda a este se debe separar por el símbolo “:”.

1.2.4.2. Documentos de tipo JSON, BSON

Los documentos JSON son de formato ligero y significa (JavaScript Object Notation – Notación de objetos JavaScript), este tipo de documentos es utilizado para realizar intercambios de datos a través de objetos JavaScript sin utilizar XML.

En cambio, BSON es un documento binario de JSON el cual esta serializado y codificado, permite la contener documentos y matrices dentro de otros documentos y matrices, también contiene extensiones que permiten la representación de otros tipos de datos que no sean JSON.

1.2.4.3. Almacenamiento orientado a documentos

MongoDB permite cambiar fácilmente el esquema de datos de cualquier aplicación, permite el almacenamiento de documentos con extensión JSON y BSON, y ofrece flexibilidad, potencia y simplicidad.

Simplicidad: Debido a que MongoDB puede almacenar datos independientemente del lenguaje de programación, y no maneja tipo de datos.

Potencia: Puede contener documentos JSON y BSON que almacenen datos de hasta 16 Mb.

Flexibilidad: Puede contener esquemas de datos dinámicos y puede suministrar varios modelos que se acoplen a la aplicación que se desee desarrollar.

1.2.4.4. Índices en MongoDB

Los índices tanto para MongoDB y las bases de datos relacionales son utilizados para realizar consultas de manera eficiente, en MongoDB los índices son usados además para restringir el numero documentos que se deben examinar, existen índices a nivel de colección, campo y también a nivel de cualquier atributo de un documento perteneciente a una colección de datos, los índices son usados para realizar condiciones en búsquedas, en la siguiente figura está representado una consulta utilizando índices en MongoDB.

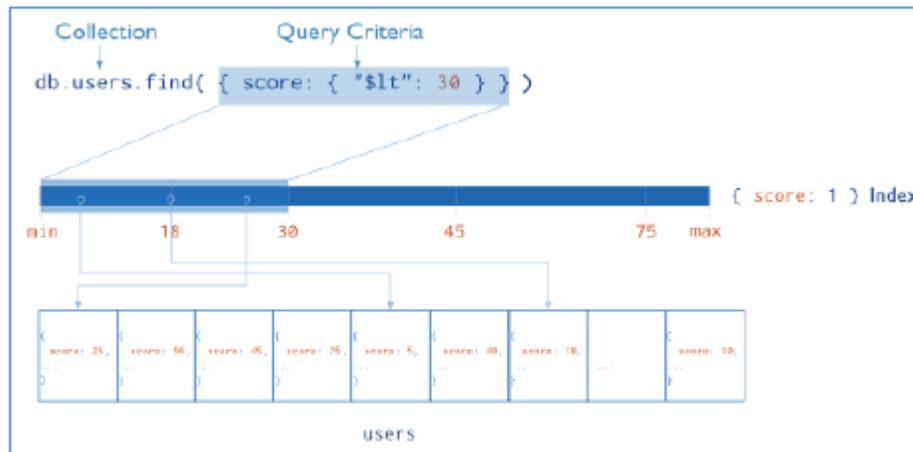


Figura 10-1: Consulta en MongoDB.

Fuente: Velastegui , Lenin, 2016.

A continuación, se describen los tipos de índices que maneja MongoDB:

- ✓ **Índices por defecto (Default_id).** Al momento de crear una nueva colección esta se asocia a una id de tipo Objeto el cual lo designa el driver de MongoDB por defecto el cual va a ser único y no se va a repetir en toda la base de datos, esto en caso que no se ingrese un valor para dicha id de antemano.
- ✓ **Único índice.** Son los índices que el usuario ingresa para poder identificar a cualquier campo de un documento.
- ✓ **Índices compuestos.** Son índices que están compuestos por dos o más campos de un documento.
- ✓ **Índices multi-índices.** Son utilizados para asociarlos con los elementos de una matriz, así como también para realizar búsquedas haciendo coincidir los elementos de las matrices.
- ✓ **Índices de texto.** Permiten guardar cadenas de texto de una colección.

1.2.5 Librería SocketIoClientDoNet

Esta es la librería que permite usar Socket.IO en Visual Studio C#, con esta librería podremos utilizar todas las ventajas que proporciona Socket.IO, y así poder conectarnos en tiempo real con el servidor que corra Node.js con Socket.IO.

Esta librería está disponible para su descarga en <https://github.com/Quobject/SocketIoClientDotNet>, o también se puede instalar directamente desde el gestor de paquetes de Visual Studio llamado NuGet.

1.3. Interfaz de usuario para pantallas táctiles

La continua evolución de la tecnología ha llevado a encontrar mejores alternativas para la comunicación hombre-máquina, es así que elementos tradicionales como el mouse, utilizado para la comunicación usuario – ordenador, va quedando relegado por el uso de pantallas táctiles, cada vez más presente en la vida cotidiana; La creación de una interfaz de usuario táctil es una alternativa acertada que mejorara la experiencia al prestar una comunicación más amigable y directa con el usuario, además agiliza los procesos al ser intuitivos y de fácil uso.

A través de una pantalla táctil permite utilizar la GUI que es la interfaz gráfica de usuario para representar gráficamente un panel de control el cual va hacer el método por el cual el usuario pueda manejar la pantalla táctil, en este caso sería por medio de toques en la pantalla o gestos realizados por los dedos del usuario sobre la pantalla.

Las pantallas y los interfaces táctiles han sido el avance definitivo para acercar más aún la informática a aquellos usuarios a los que el ratón y teclado les generaban temor e inseguridad. El poder tocar una pantalla directamente con sus dedos hacía que perdieran el miedo a la informática.

1.4. Metodología SCRUM

La metodología ágil de SCRUM está ligado a la utilización mejores técnicas de trabajo en equipo donde el entendimiento entre el equipo de desarrollo y el usuario es prioritario para obtener resultados positivos en el proyecto.

Las técnicas propuestas por SCRUM estas relacionadas unas con otras con el fin de dar resultados altamente beneficiosos para el desarrollo del sistema. Permite realizar entrega periódicamente de los avances del proyecto, manteniendo informado al usuario durante el desarrollo del proyecto.

Además, Scrum está diseñado para proyectos grandes medianos y pequeños, en los cuales la obtención de resultados son prioritarios y los requerimientos por parte del usuarios pueden ser cambiantes.

1.4.1. Características de SCRUM

SCRUM establece un modelo a seguir durante el desarrollo de todo el proyecto, es el punto de partir el cual regirá las actividades que se realizaran a lo largo del proyecto. Se deben establecer roles dentro de los denominados “teams” que son los equipos de trabajo.

Lo primero que se debe definir antes de empezar el trabajo es el Product-Backlog que es una representación de los requisitos que definieron conjuntamente con el cliente al inicio del proyecto, estas actividades a realizar deben estar priorizadas, a todo este conjunto de actividades se las suele dividir en “sprints”, es decir a lo largo del proyecto dependiendo del tamaño del mismo, se va a tener algunos sprints que normalmente duran entre 1 – 4 semanas, cada sprint debe arrojar como producto un entregable, es decir un porción del producto final, lógicamente este debe ser utilizable.

Dentro de la metodología se contempla también lo que es el Sprint Planning, esta es una reunión en la que se definen que elementos del Product-Backlog forman cada sprint, además en esta reunión de dueño del producto determina que elementos desea ver terminados.

El proceso de desarrollo mediante la metodología SCRUM, contiene las siguientes actividades:

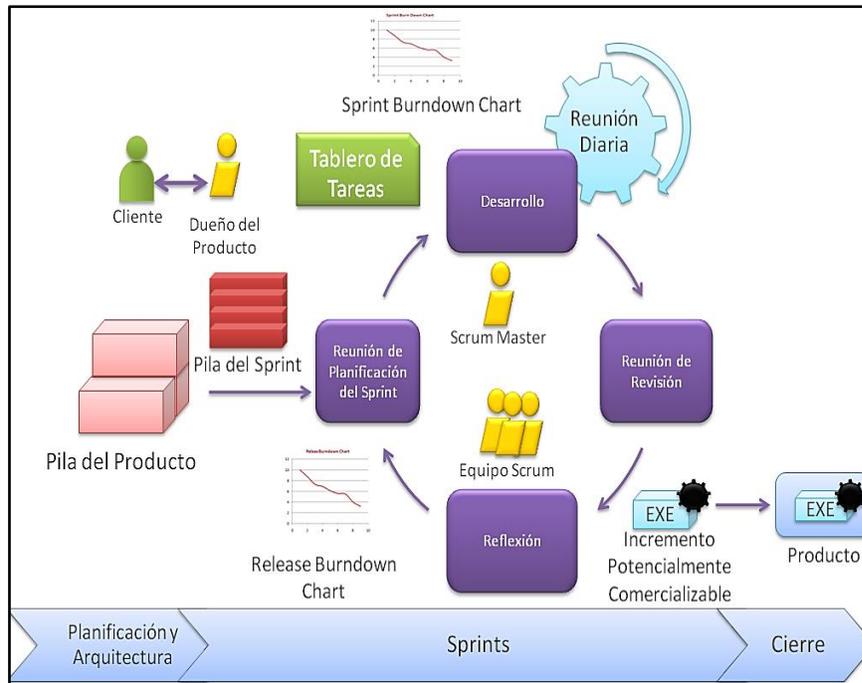


Figura 11-1: Procesos de Scrum.

Fuente: <http://etherpad.proyectolatin.org/up/aaea7098b048d17f4bbe2a2bb599abeb.png>

1.4.2. Roles de SCRUM

Como lo veníamos mencionando SCRUM organiza a un equipo de trabajo, que se conoce como team y dentro del cual lógicamente se definen roles. Dentro de los principales roles tenemos:

ScrumMaster

Más que un jefe es un líder, es el encargado de guiar al equipo y eliminar las barreras que impidan que el equipo complete satisfactoriamente cada sprint.

Product Owner

Es el encargado de velar por los intereses del cliente, este es el encargado de crear las historias de usuario, priorizarlas y colocarlas dentro del Product Backlog.

Team

Es el equipo de trabajo encargado de ejecutar las historias de usuario, normalmente establecido por grupos de 3 – 8 personas, son los encargados de entregar los productos.

Stakeholders

Normalmente es el cliente propiamente dicho, es para quien se está desarrollando el proyecto, únicamente participan en lo que son las revisiones de cada sprint.

1.4.3. Ciclo de vida de SCRUM

Este ciclo se encuentra conformado por 4 etapas o fases, las cuales se describen a continuación:

Fase de Planificación

En esta fase se define todo el trabajo que se va a realizar, además de los equipos de trabajo y los roles de cada miembro dentro de cada equipo.

Fase de Diseño

En esta fase se definen normalmente los detalles técnicos del proyecto: como lo son bases de datos, IDE's, Frameworks, versionadores de código y demás herramientas necesarias para el proyecto.

Fase de Codificación

Se desarrollan los Sprint de cada iteración, es aquí donde realmente empieza a convivir el entorno de trabajo del proyecto, cada entregable al final del proyecto se convierte en el producto final que se entrega a los stakeholders.

Fase de Pruebas

Normalmente se suele realizar antes de la entrega de cada sprint, aunque al final del proyecto se puede hacer una revisión general del producto a entregar.

1.4.4. SCRUM vs metodologías tradicionales

Mientras que las metodologías tradicionales se basan en un lineamiento fijo en cual se planifican todas las actividades de inicio a fin, a las cuales las subdivide en etapas, tales como: recolección de requerimientos, análisis de procesos, diseño, desarrollo, realización de pruebas y entrega del producto final. En cambio, SCRUM se centra en realizar pequeñas interacciones las cuales se van entregando al terminar cada actividad permitiendo al usuario final observar el progreso que va teniendo el producto, de tal forma que no espera hasta el final del proyecto.

Las metodologías tradicionales presentan dificultades ante el cambio, ya que mientras más este avanzado el proyecto es más difícil realizar ajustes. SCRUM está diseñado para permitir cambios en el transcurso del desarrollo de software ya que, con iteraciones cortas, el optimizar o modificar una característica del proyecto no conlleva mucho trabajo, tan solo se debe dar prioridad al cambio e incluirlo en la iteración a la que pertenezca.

Así también, las metodologías tradicionales no involucran de forma permanente al usuario, el cual interviene en la recolección de requerimientos y permanece ausente durante las siguientes etapas de diseño, construcción, pruebas y nuevamente interviene ya al final en la entrega del producto. Mientras que con SCRUM la intervención del usuario puede ser permanente para cada una de las etapas en las cuales el usuario final puede observar la evolución del sistema y también opinar en cómo se puede mejorar.

Las metodologías tradicionales centran su atención en cómo va el avance del proyecto en relación a lo planificado, tanto en las actividades como en el software final. SCRUM centra su atención en si el usuario se siente satisfecho con las versiones del sistema que se le va entregando y caso contrario en cómo se puede mejorar cada interacción hasta obtener la aprobación del cliente.

Un ejemplo práctico de las problemáticas que presentan las metodologías tradicionales sería que en un X proyecto que se está desarrollando en el cual se reemplazó al encargado de realizar el análisis y recolección de requerimientos, y el nuevo integrante no le convence algunas definiciones iniciales y solicita cambiarlas sin tomar en cuenta que el software ya se encuentra en una fase avanzada de desarrollo, los cambios no se le notifica al usuario final y no es hasta el final que se nota la insatisfacción del cliente . ¿Entonces como la metodología ágil SCRUM soluciona este problema? En cada ciclo de iteración existe el *Sprint Review* que es la actividad que pretende involucrar al usuario al final de cada interacción para que observe el avance del proyecto y de su opinión al

respecto; en este punto, todos los involucrados pueden participar, de esta manera el usuario que recibía el sistema debe entender que era una decisión en conjunto y no de una sola persona.

1.4.5. SCRUM vs otras metodologías ágiles

SCRUM cumple con todos los principios básicos de las metodologías ágiles tales como: entrega de versiones del producto, constante retroalimentación por parte del cliente y los integrantes del equipo de trabajo, permite cambios cuando se amerite y una buena comunicación entre el cliente y los desarrolladores. Sin embargo, ya entrando al detalle, tiene diferencias importantes con otras metodologías ágiles que vale la pena destacar.

Para esta comparativa se tomó en cuenta a la metodología Xtreme Programming “XP” que es una de las más importantes dentro de la categoría de las metodologías ágiles.

- **El tiempo:** En XP, el rango en tiempos es de una o dos semanas y es menor al tiempo que ocupa SCRUM que es de dos a cuatro semanas, estos son los tiempos estándares para cada metodología, pero los tiempos se pueden acoplar de acuerdo a las necesidades del proyecto.
- **El proceso:** SCRUM no imponen las herramientas que se deben usar en la metodología, es el desarrollador el que escoge lo que debe usar. En cambio, en otras metodologías ágiles como la mencionada XP define herramientas que debe usar como parte de su estándar, por ejemplo, desarrollo orientados a pruebas.
- **Los cambios:** SCRUM recomienda no realizar cambios durante el desarrollo de la interacción, en cambio otras metodologías son más flexibles y si permiten realizar estos cambios.
- **El orden:** En SCRUM es el equipo de trabajo quien ordena la secuencia de las interacciones basándose en las necesidades del cliente, en XP es el cliente quien define el lineamiento de las interacciones.

- Scrum se permite que el equipo con base en la prioridad definida por el cliente sea quién decida en qué se puede comprometer para cada iteración en cuanto a desarrollo se refiere, en XP el equipo debe seguir el lineamiento dado por el cliente.
- **La retroalimentación:** Scrum recomienda realizar retroalimentación al final de cada sprint , en XP la retroalimentación es temprana a medida que se desarrolla la entrega.

1.4.6. Ventajas de SCRUM

Todas las metodologías tienen puntos positivos que justifican su utilización, por eso es importante analizar que metodología utilizar basándose en el tipo y las necesidades del software que se va a desarrollar. El siguiente listado contiene algunas de las ventajas de SCRUM sobre otras metodologías.

- **Satisfacción del cliente:** La gran diferencia entre metodologías ágiles y tradicionales, es que en las ágiles hacen que el cliente pase a formar parte del equipo de trabajo, de esta manera el usuario puede intervenir en cualquier fase de desarrollo de software y dar su opinión comprometiéndole así con el resultado final. En las metodologías tradicionales, el cliente solo interviene para la recolección de datos, posteriormente aprueba extensos documentos y no volverá intervenir hasta que el producto ya esté terminado.
- **Simplicidad:** los eventos manejados por Scrum están clara-mente identificados, indicando para cada uno: quienes participan, su objetivo, el tiempo que debe tomar y cuál es el resultado esperado. Lo cual en esencia facilita a los integrantes del equipo la adopción de la metodología.
- **Inspección:** uno de los componentes que resalta Scrum, es la inspección y por ello, tres de sus eventos están orientados a estos objetivos: la reunión diaria, la revisión del sprint y la retrospectiva de este último. Estos eventos permiten a la organización consolidar la metodología y detectar en cada equipo y en cada proceso, qué debe ser mejorado. Este componente es uno de los preferidos en las organizaciones, ya que les permite ver qué tan bien se va adaptando la metodología a su cultura y si los beneficios prometidos se están evidenciando.
- **Adaptación:** la mejor parte de la metodología es la disposición que tienen al cambio las características del producto. Este es uno de los componentes que más la diferencia con el resto,

ya que el cambio puede ser efectuado en cualquier momento, incluso dentro del desarrollo de la ejecución de las diferentes iteraciones o *Sprints* siempre y cuando no afecte la entrega pactada. Esta adaptación beneficia a la organización en la medida que aporta a la satisfacción del cliente y los ingresos por ajustes.

- **Trabajo en equipo:** algo particularmente interesante de Scrum es cómo logra la sinergia entre las personas que participan en el proceso, a tal punto que en cada iteración -ciclo de desarrollo-, el mismo equipo se adapta para mejorar. Esto también implica que cada individuo sea reconocido como parte esencial del equipo, por lo cual el impacto del cambio de una persona puede llegar a ser alto. Comparativamente, Scrum logra dar visibilidad al equipo de trabajo, ya que, en metodologías tradicionales, las personas que lo integraban no tenían relación directa con el cliente, lo cual, en términos de reconocimiento, es importante y vale la pena destacarlo.

1.4.7. Limitaciones de SCRUM

Como toda metodología, Scrum a pesar de las bondades que presenta, también tiene limitaciones que deben tenerse en cuenta a la hora de ser implementado:

- **Complejidad en su implementación:** para equipos que vienen del desarrollo tradicional enfrentarse a ideas tales como ser auto-gestionados, estar abierto a cambios que pida el cliente, o estar en contacto directo con el cliente, puede ser todo un reto y requiere ejercicios previos antes de ponerse en práctica.
- **Tiempo por parte del cliente:** esta metodología requiere tiempo considerable por parte del cliente y contacto permanente con el equipo, por lo cual puede que no en todos los escenarios se cuente con la disponibilidad de trabajar de esta forma.
- **Contratos costo y alcance definido:** aunque es posible trabajar con Scrum en este tipo de contratos, se pierde algo básico de la esencia y es la adaptabilidad, el poder cambiar o mejorar características del sistema a medida que evoluciona el producto. Lo anterior. Considerando que se tiene una fecha objetivo para cumplir con las características bajo las cuales se hizo la contratación.

- **Documentación requerida:** parte del objetivo de las metodologías ágiles, es documentar lo estrictamente necesario, pero ¿Qué pasa cuando el cliente es el que exige documentación exhaustiva y prototipos detallados previos al desarrollo? En estos escenarios Scrum no luce tan bien, pues atenta contra la base fundamental que es el agilísimo.
- **Stress:** este es un fenómeno que se da no solo hablando de Scrum sino en general de cualquier metodología de desarrollo ágil, y es que el equipo, en aras de cumplir con las entregas en ciclos cortos que se repiten indefinidamente, cae en una etapa de *stress*, ya que los miembros siempre están pensando en compromisos, tiempo y estándares de desarrollo.
- **Proyecto de gran envergadura:** Scrum está pensado para un equipo de hasta nueve personas, si más de esto es requerido, se debe pensar en varios equipos de Scrum, lo cual a su vez, implicaría una coordinación y un seguimiento que puede no ser fácilmente aplicable en el proyecto.
- **Ingenieros juniors:** debido al aprendizaje y sinergia que logra el equipo, la inclusión de ingenieros con menos experiencia o sin experiencia en la metodología, implica un acompañamiento muy de cerca que puede llegar a comprometer los tiempos que se tienen para el desarrollo del producto.

CAPITULO II

2. MARCO METODOLOGICO

Para llevar una correcta gestión del desarrollo del proyecto es necesario la utilización de una metodología de desarrollo de software se ha optado por la metodología ágil SCRUM puesto que estas permiten la intervención directa del usuario durante el desarrollo del software, ofrecen una rápida respuesta a cambios de requisitos gracias a su proceso iterativo, permitiendo el desarrollo, testeo y correcciones rápidas que facilita la entrega de productos de calidad a tiempo.

2.1. Fase de planificación

Como su nombre lo dice en esta etapa del desarrollo de software se realiza la planificación para el desarrollo del sistema de control de acceso de estudiantes para el CTE de la UNACH, esto gracias a la recolección de los requerimientos de usuarios, los mismo que se extraen de las reuniones y/o entrevistas a los técnicos del CTE.

El proyecto está sujeto al siguiente diagrama de actividades fig. 1-2, tomando en cuenta que los tiempos para cada actividad pueden cambiar durante la realización del sistema.

	i	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Pred
1			FASE DE INVESTIGACION	25 días	lun 25/04/16	vie 27/05/16	
2			Recolección de informacion	10 días	lun 25/04/16	vie 06/05/16	
3			Consulta Bibliografica	7 días	lun 09/05/16	mar 17/05/16	2
4			Analisis de informacion	8 días	mié 18/05/16	vie 27/05/16	3
5			FASE DE EJECUCIÓN	80 días	lun 30/05/16	vie 16/09/16	1
6			Diseño Procidemental	4 días	lun 30/05/16	jue 02/06/16	1
7			Arquitectura de software	5 días	vie 03/06/16	jue 09/06/16	6
8			Diseño de interfaz	10 días	vie 10/06/16	jue 23/06/16	7
9			Codificación	40 días	vie 24/06/16	jue 18/08/16	8
10			Pruebas	5 días	vie 19/08/16	jue 25/08/16	9
11			Correcciones	16 días	vie 26/08/16	vie 16/09/16	10
12			FASE DE DOCUMENTACIÓN	30 días	lun 19/09/16	vie 28/10/16	11
13			Redacción de borrador de trabajo de titulación	10 días	lun 19/09/16	vie 30/09/16	11
14			Documentacion del trabajo de titulación	12 días	lun 03/10/16	mar 18/10/16	13
15			Tramites y presentación del trabajo de titulación	8 días	mié 19/10/16	vie 28/10/16	14

Figura 1-2: Actividades Planificadas

Fuente: Velastegui , Lenin, 2016.

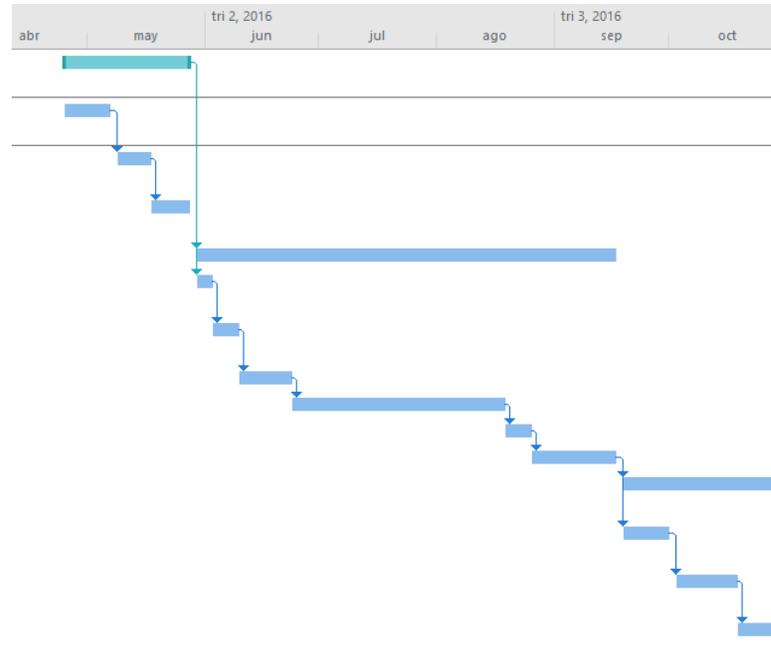


Figura 2-2: Diagrama Gantt de actividades

Fuente: Velastegui , Lenin, 2016.

2.1.1. Reuniones

En las distintas reuniones que se realizaron con los técnicos y directivos del CTE se recolectaron los requerimientos, en la siguiente tabla se muestra un resume de lo acordado en cada reunión que se realizó.

Tabla 1 – 2: Reuniones fase de planificación

Fecha	Asistentes	Actividades	Resultado
25 – 04 – 2016 Al 28-04-2016	Director del CTE – Universidad Nacional de Chimborazo Ing. Jorge Delgado Tec. Patricio Ochoa. Lenin Velastegui O.	Recopilación de información: proceso de ingreso de los estudiantes al laboratorio de internet.	Establecimiento de los módulos del sistema
29 – 04 – 2016	Director del CTE – Universidad Nacional de Chimborazo Ing. Jorge Delgado Tec. Patricio Ochoa. Lenin Velastegui O.	Definición de los requerimientos de usuario	Creación del listado de requerimientos del sistema
06 – 05 – 2016	Director del CTE – Universidad Nacional de Chimborazo Ing. Jorge Delgado Tec. Patricio Ochoa. Lenin Velastegui O.	Presentación de los módulos y requisitos del sistema	Aceptación de requisitos del software

Fuente: Velastegui , Lenin, 2016.

Luego de las reuniones se estableció la creación de cuatro módulos:

Para una mejor organización el sistema estará dividido en módulos, que permitirán gestionar la información, los mismos que estarán definidos de la siguiente forma:

Módulo de cliente. - Este módulo permitirá contendrá las acciones que puede realizar el cliente dentro del sistema.

Las tareas a realizar en este módulo son:

El cliente pueda escoger un determinado computador.

El cliente pueda entrar a la opción de asignación aleatoria de un computador.

Ingresar los datos de registro de uso de las pcs que contendrán los siguientes datos (id del usuario, fecha y hora del inicio de sesión, tiempo de uso, numero de computador, fecha y hora de la finalización del uso del computador).

Módulo de Administrador. - Este módulo contendrá las acciones que puede realizar el administrador dentro del sistema.

Las tareas a realizar en este módulo son:

Tiempo: Que permitirá asignar un tiempo de uso sin restricciones a todos los computadores clientes.

Uso Libre: Así como la acción anterior, pero en este caso el administrador no dará un tiempo, él es libre de activar o desactivar el uso sin restricciones de los clientes.

Consultas: permitirá generar consultas de datos que se encuentren almacenados en el sistema. por ejemplo: lista de usuarios bloqueados que no pueden ingresar al sistema.

Acciones directas con los clientes: permitirá apagar o reiniciar los computadores de la sala de internet.

Bloquear y desbloquear a usuario para el uso de los computadores.

Módulo de autenticación y control. - Este módulo permite la auto identificación de las personas que deseen ingresar a la sala de internet.

Módulo de reportes. - Este módulo permitirá generar reportes de los principales procedimientos que realizará el sistema de gestión de control de acceso además de los reportes requeridos por el usuario.

2.1.2. *Procesos a automatizar*

A continuación, se exponen los procesos que se pretenden automatizar con la realización del sistema de control de acceso.

- Ingreso de los estudiantes al laboratorio del CTE: El estudiante ingresara de una forma más eficiente al laboratorio debido a que ya no estará proporcionando sus datos completos de forma manual a los custodios del departamento, así se evitaran largas colas al momento del registro de uso, ahora el estudiante tan solo proporcionara su usuario y contraseña en la maquina principal y después que el sistema verifique los datos podrá ingresar al sistema para poder escoger el computador que desee de una lista de pcs disponibles.
- Elección de Pc: El estudiante tendrá dos opciones de escoger el computador que desee, la primera opción es desde la maquina central (Administrador) y la segunda opción es acercarse directamente al computador (cliente) que no se esté utilizando he ingresar su usuario, contraseña y así poder utilizar la pc sin pasar primero por la maquina central.
- Registro del uso de los computadores del laboratorio del CTE: Al momento de validar el usuario y contraseña del estudiante, el sistema automáticamente registrara la fecha y la hora del inicio de la utilización del computador elegido, el id del usuario que está usando el computador, el número de máquina que es el identificador del pc que ha escogido el estudiante, y así también al momento de cerrar sección desde el cliente se guardara la fecha y hora de la finalización del uso del computador.
- Reporte de estudiantes: Se puede generar reportes de los estudiantes que han ocupado los computadores desde un rango de fechas ingresadas por la persona administradora.
- Reporte de estudiantes bloqueados: Se puede generar un listado de los estuantes que no puedan ingresar al sistema.
- Permite automatizar el proceso de reinicio y apagado de todos los computadores, de esta forma la persona encarga del laboratorio no tendrá que estar apagando o reiniciando las pcs una por una.

2.1.3. *Personas y roles del proyecto*

Para el presente proyecto se han definido los siguientes roles:

Tabla 2 – 2: Roles del sistema

Rol	Nombre
Product Owner	Técnico del CTE: Sr. Patricio Ochoa
Stakeholders	Director del CTE: Ing. Jorge Delgado
ScrumMaster	Dr. Julio Santillán.
Team	Lenin Velastegui O.

Fuente: Velastegui , Lenin, 2016.

2.1.4. *Pila del producto*

El Product Backlog o Pila del producto representa al conjunto de historias de usuario que conforman el sistema. En la Tabla 3 – 2: Pila del producto, se detallan todas las historias de usuario.

Tabla 3 – 2: Pila del producto

Pila del Producto	
ID	Descripción
1	Como programador deseo obtener los requerimientos del sistema proporcionados por el usuario.
2	Como técnico deseo diseñar el modelo de la arquitectura del sistema.
3	Como programador deseo especificar un estándar de codificación.
4	Como programador deseo diseñar la base de datos para el sistema.
5	Cómo programador deseo crear las clases necesarias para correcto funcionamiento del sistema.
6	Cómo programador deseo obtener el diseño de la interfaz de usuario para el sistema.
7	Como administrador deseo poder ingresar al sistema a través de un username y contraseña.
8	Como administrador deseo digitar mi usuario y contraseña de forma táctil.

9	Como estudiante deseo digitar mi usuario y contraseña de forma táctil.
10	Como administrador y/o estudiante deseo tener una pantalla principal donde se muestren las opciones de ingreso al sistema.
11	Como administrador deseo tener una interfaz donde se muestren las opciones que pueda realizar.
12	Como administrador deseo poder desbloquear todos los computadores por un determinado tiempo ingresado de antemano.
13	Como administrador deseo poder desbloquear todos los computadores hasta cuando elija bloquearlos.
14	Como Administrador deseo obtener una lista de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).
15	Como Administrador deseo obtener una lista de los estudiantes que no pueden ingresar al sistema.
16	Como administrador deseo poder bloquear o desbloquear el ingreso al sistema.
17	Como administrador deseo reiniciar todos los computadores que estén conectados a la maquina principal.
18	Como administrador deseo apagar todos los computadores que estén conectados a la maquina principal.
19	Como administrador deseo cerrar sesión en el sistema y volver a la pantalla de inicio.
20	Como estudiante deseo poder ingresar al sistema después de autenticarme.
21	Como estudiante deseo escoger un computador que esté disponible.
22	Como estudiante deseo que el sistema me asigne un computador de forma aleatoria de los que se encuentren disponibles.
23	Como estudiante deseo cerrar sesión en el sistema.
24	Como estudiante deseo tener un menú donde se indique la hora de inicio y tiempo de uso del computador, además de poder cerrar sesión.
25	Como estudiante deseo poder desbloquear desde el mismo computador cliente sin tener que ir a la maquina central.
26	El sistema debe actualizar y mostrar automáticamente las pcs que estén disponibles
27	La app cliente y administrador deben conectarse automáticamente al servidor
28	Generar un reporte de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).
29	Generar un reporte de los estudiantes que no pueden ingresar al sistema.

30	El sistema debe retirar de la lista de pcs disponibles cuando la app cliente se desconecte del servidor
31	Como estudiante deseo que el sistema guarde automáticamente los datos de uso del pc.

Fuente: Velastegui , Lenin, 2016.

El proyecto consta de 31 historias de usuario las cuales serán organizados en varios sprints, mismos que al final del desarrollo conformarán un producto entregable.

2.1.5. *Análisis económico*

Para el desarrollo e implementación del Sistema informático para la gestión de la información del control de acceso de los estudiantes a la sala de internet del CTE no fue necesario realizar ningún tipo de gastos de implementación ni gastos de licencia ya que el CTE dispone del hardware y software necesario tanto para el desarrollo como para la implementación y puesta en funcionamiento de la aplicación.

Tabla 4 – 2: Presupuesto del proyecto

ITEMS	CANTIDAD	VALOR UNITARIO	TOTAL
SOFTWARE			
Visual Studio 2012 ultimate	1	369,50	369,50
MongoDb	1	0	0
Node.js	1	0	0
Socket.io	1	0	0
Office 2016 profesional	1	119,99	119,99
WebStorm 2016 durante 1 año	1	129	129
HARDWARE			
Computador All in One Pantalla Tactil Intel core i7 3.2 GHz, Memoria RAM 8 Gb, disco duro de 1 TB. (maquina central).	1	1250	1250

Computador de escritorio Intel core i7 3.2 GHz, Memoria RAM 8 Gb, disco duro de 500 Gb. (maquina cleinte).	278	855	237.690
Laptop Asus Intel core i7 3.2 GHz, memoria Ram 8 Gb, disco duro 1Tb	1	1015	1015
Impresora Multifunción	1	230	230
OTROS			
Bibliografía	Varios	100	100
Suministros de Oficina	Varios	50	50
Internet	6 meses	20	120
Transporte y Viáticos	6 meses	50	300
Servicios Básico	6 meses	30	180
TOTAL			241.553,49

Fuente: Velastegui , Lenin, 2016.

El costo del desarrollo del proyecto tendrá un valor total de \$ 241.553,49 , el cual el autor del proyecto cubrirá con \$ 1995 y el CTE aportara con las licencias del software a emplear y los dispositivos hardware para la implementación del sistema los cuales tienen un valor de \$239.558,49.

2.2. Fase de diseño

En este punto se definen actividades previas iniciales antes de empezar con el desarrollo de las tareas como, por ejemplo: la arquitectura de la aplicación, los recursos necesarios para el desarrollo del sistema, la definición de un estándar de codificación, diseño de la GUI, el diseño del BD y el diagrama de clases.

2.2.1. Diagrama de Procesos

Para tener una visión del proyecto se debe representar por medio de un diagrama en el cual se refleja cada actor y los procesos que intervienen en el sistema.

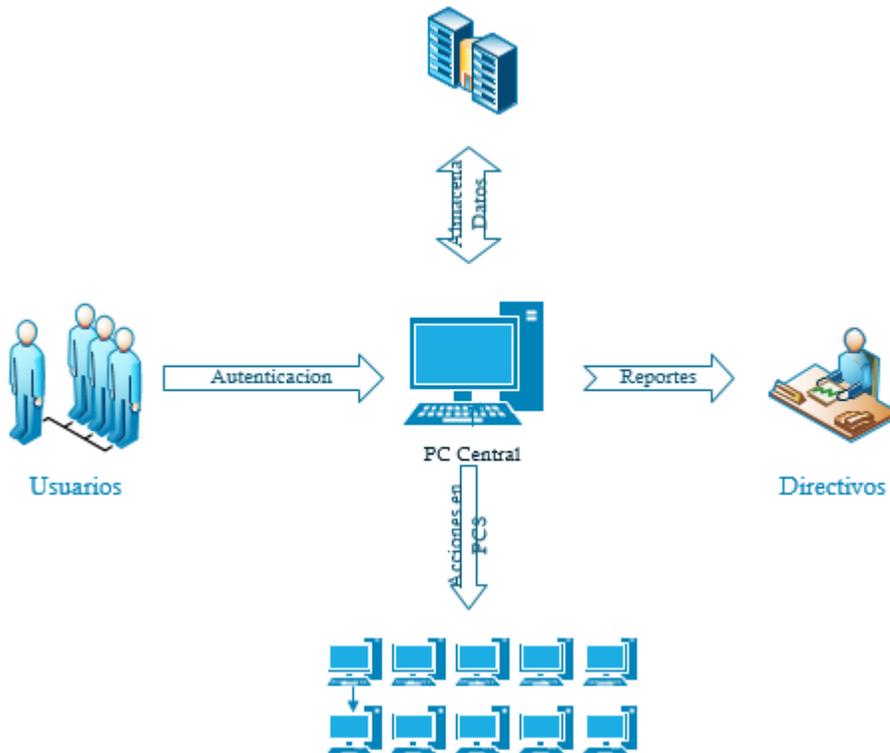


Figura 3–2: Diagrama de Procesos.

Fuente: Velastegui , Lenin, 2016.

2.2.2. Diagrama de casos de uso

Los diagramas de caso de uso permiten representar la relación que existe entre los usuarios y los procesos del sistema, en la siguiente figura se representa la interacción entre el usuario administrador y estudiantes con relación a sus respectivos procesos.

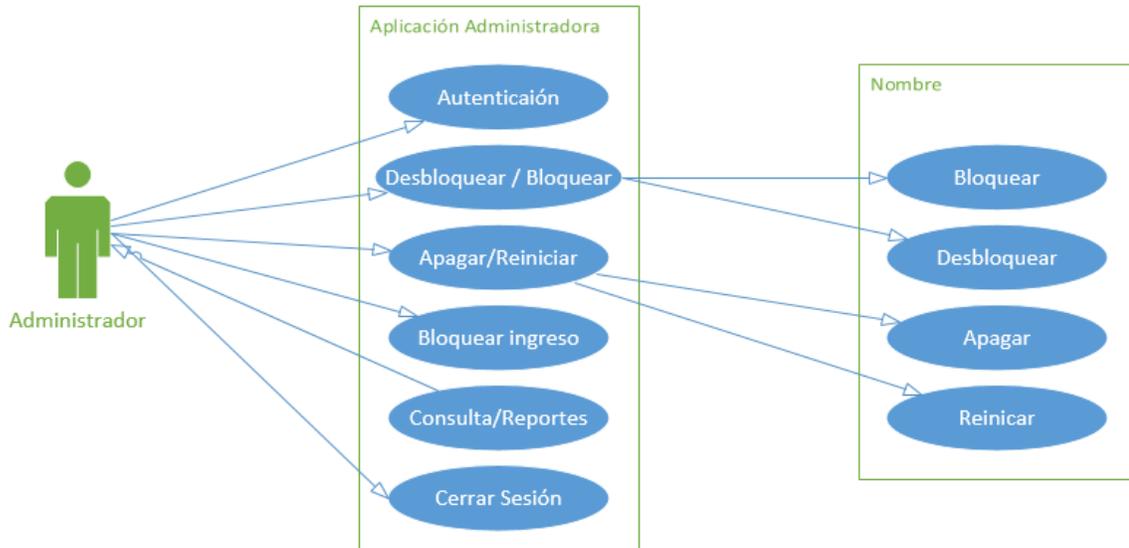


Figura 4–2: Diagrama de casos de uso administrador.

Fuente: Velastegui , Lenin, 2016.

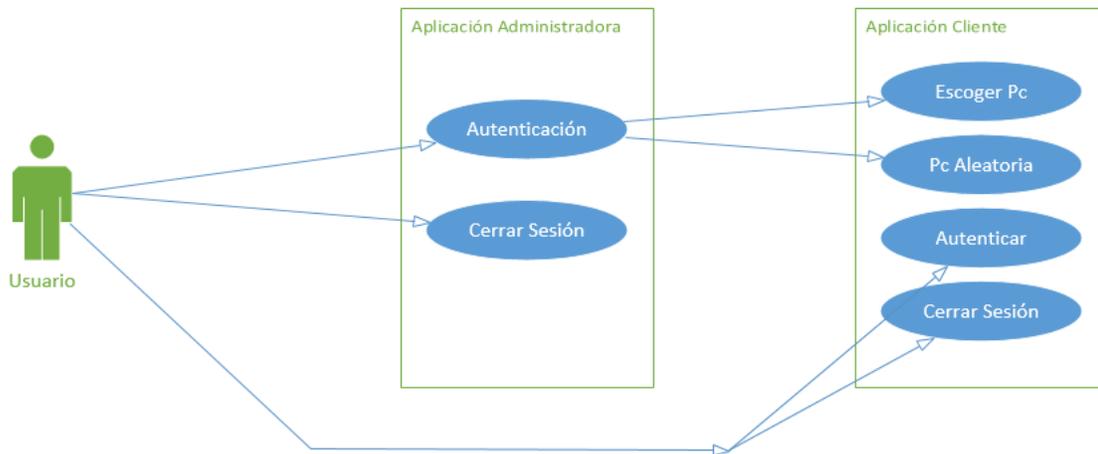


Figura 5–2: Diagrama de casos de uso cliente.

Fuente: Velastegui , Lenin, 2016.

Existen dos diagramas de casos de uso en el proyecto debido a que existe una aplicación para el administrador y otra aplicación para los clientes.

2.2.3 Diagrama de clases

Este diagrama contiene las clases necesarias para el correcto funcionamiento del sistema y así cumplir con los procesos anteriormente descritos.

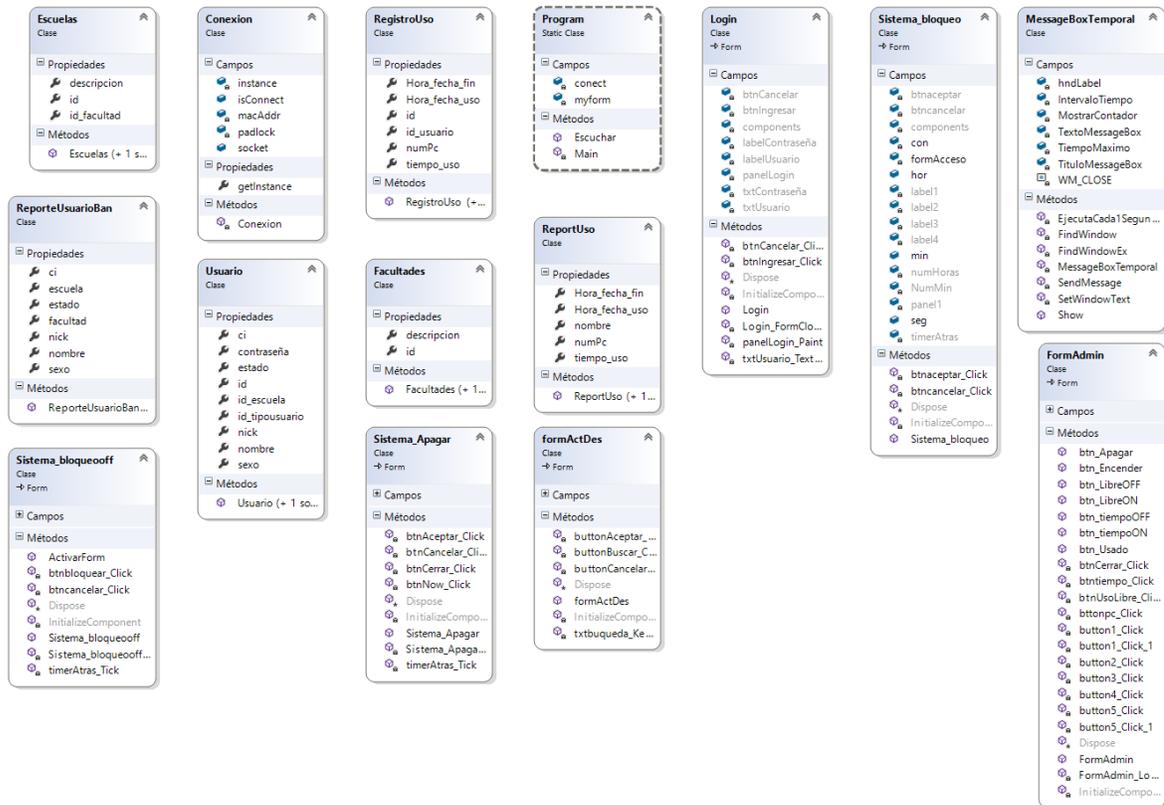


Figura 6–2: Diagrama de clases administrador

Fuente: Velastegui , Lenin, 2016.

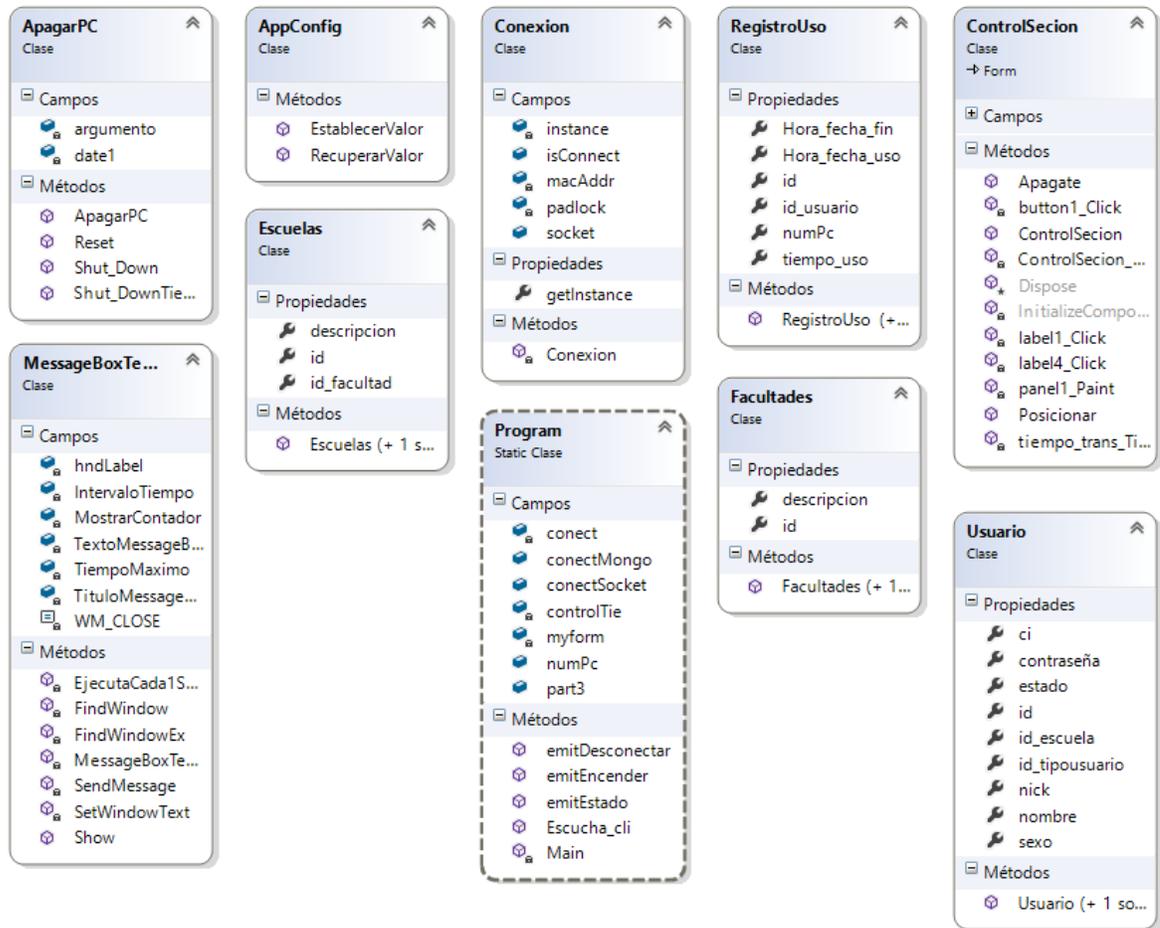


Figura 7–2: Diagrama de clases cliente

Fuente: Velastegui , Lenin, 2016.

Existen 2 diagramas de clases en el proyecto debido a que existe una aplicación para el administrador y otra aplicación para los clientes.

2.2.4. *Arquitectura de la aplicación*

La creciente demanda de sistemas basados en entornos web ha marcado una tendencia en la industria, sin embargo, para este caso se optó por el desarrollo de un sistema de escritorio. Debido a las condiciones del negocio se puede determinar una arquitectura para un determinado proyecto, que para este caso es una arquitectura cliente servidor, la misma que responde al escenario del proyecto de una manera satisfactoria.

La arquitectura de la aplicación para el presente proyecto está dividida en tres capas, las cuales se detallan a continuación:

- Capa de acceso a datos: En esta capa se definen todas las clases y objetos necesarios para la conexión a la base de datos desde la aplicación. Es decir, es por aquí deben pasar todos los datos tanto para la lectura como para la escritura.
- Capa de lógica del negocio: Esta capa es la encargada del procesamiento de la información, misma que deben cumplir con las políticas del negocio.
- Capa de presentación: En esta capa se definen las interfaces de usuario, mismas que interactúan con el usuario para la presentación y/o lectura de los datos.

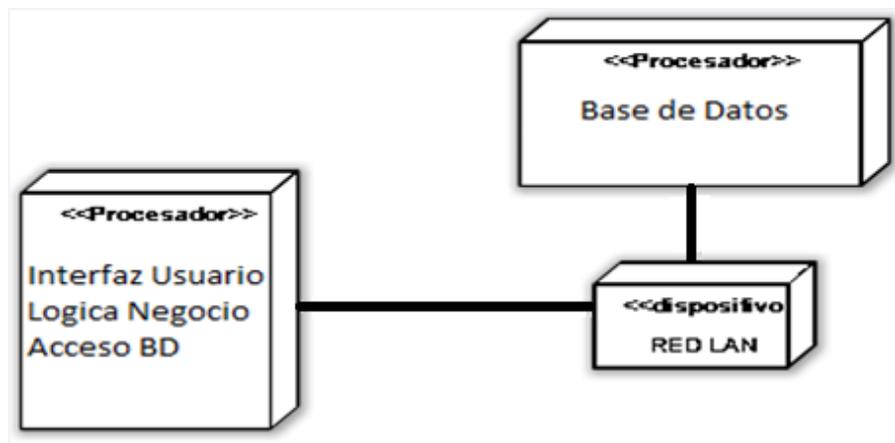


Figura 8–2: Arquitectura del sistema.

Fuente: Velastegui , Lenin, 2016.

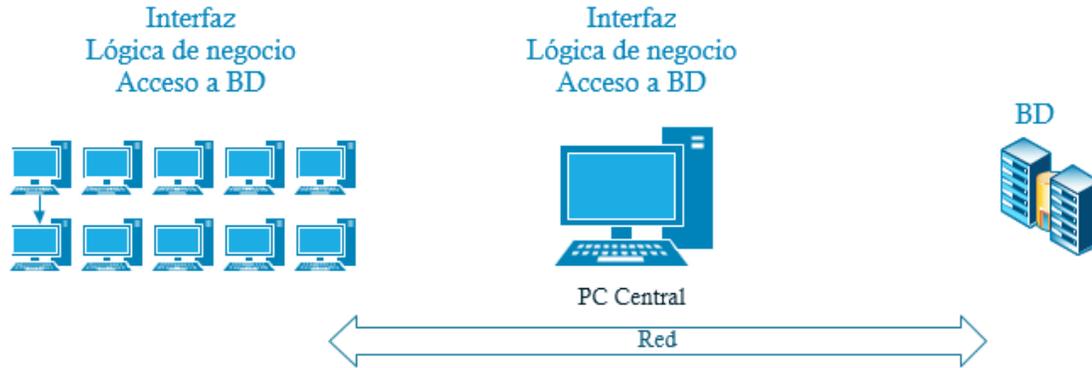


Figura 9–2: Arquitectura aplicada al sistema de control de accesos.

Fuente: Velastegui , Lenin, 2016.

2.2.5. Recursos necesarios

Los siguientes recursos hardware, software y materiales, fueron utilizados:

Tabla 5–2: Recursos necesarios

TIPO DE RECURSO	DESCRIPCION
Hardware	Laptop Intel Core i7, 3.1 GHz, Disco duro de 1 Tb, 8Gb. RAM
Software	Windows 10 Pro. Visual Studio 2012. MongoDB 3.2.5 Node.js 4.4.3 Socket.io 1.4.5 Microsoft Office
Materiales	Flash Memory Disco Duro Internet

Fuente: Velastegui , Lenin, 2016.

2.2.6. Estándar de codificación

Para los nombres de las clases de cada paquete, objetos, métodos correspondientes, estos deben ser de acuerdo a la función que van a cumplir.

Tabla 6 – 2: Estándar de codificación

COMPONENTE	NOMBRE	DESCRIPCIÓN
Clases	<Nombreclase> <NombreclaseUnoDos>.	La primera letra con mayúscula seguida de letras minúsculas para el nombre de la clase, en caso de ser varias palabras las primeras letras de cada palabra con mayúscula.
Variables	<cont>	Las variables a usar serán declaradas en minúsculas
Objetos declarados	obj_<Nombreobjeto>	Los objetos declarados se los escribirá con el prefijo obj
Paquetes	Acceso_Datos	Los nombres de los paquetes tendrán será 1 a primera letra con mayúscula y las demás seguidas de minúsculas separadas por un guión bajo
Botones	btn<Nombre button>	Empieza con el prefijo btn
Label	label<Numero>.	Empieza con el prefijo label
TextBox	txt<Nombre TextBox>.	deben llevar el prefijo txt
combo box	combo<Nombre combo box>.	Inicia con el prefijo combo
Panel	panel<Numero>.	Prefijo panel y numero

Fuente: Velastegui , Lenin, 2016.

2.2.7. Diseño de la interfaz de usuario

Una interfaz amigable es uno de los pilares fundamentales para que un sistema sea aceptado por un usuario, se utilizaron colores claros una mezcla entre azul blanco y colores tomados desde la página oficial de la UNACH.

Una vez implementado se realizó pruebas de aceptación con el usuario de visibilidad teniendo como resultado que no existiría problemas y las interfaces con esos colores no provocan cansancio al momento de utilizarlas por tiempos prolongados.

En la figura 10–2 podemos observar la interfaz de usuario con las especificaciones antes mencionadas.

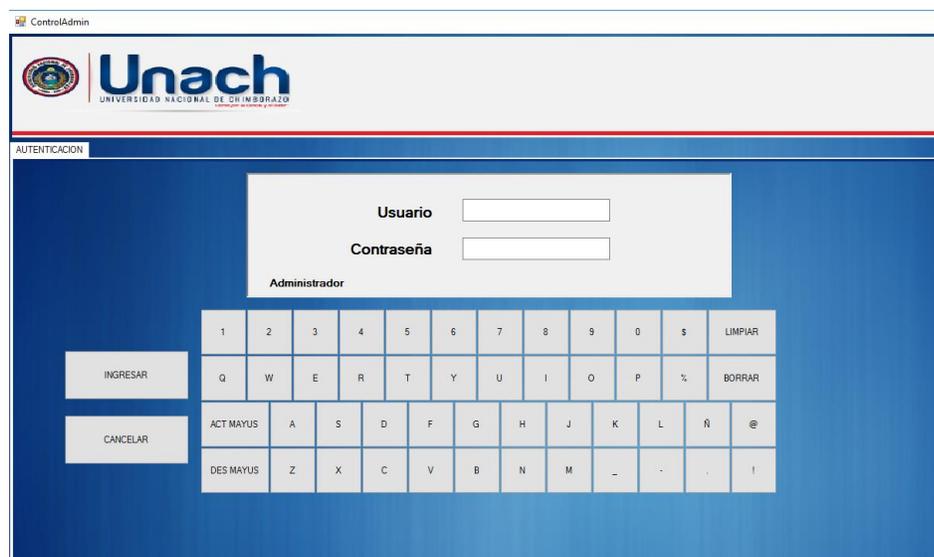


Figura 10–2: Interfaz del sistema – Pantalla de login

Fuente: Velastegui , Lenin, 2016.

Módulo de control y autenticación

Pantallas principales app administrador y cliente:

El módulo de control y autenticación nos permite mostrar las pantallas principales tanto al administrador como al cliente, las cuales están totalmente bloqueando las pantallas de los computadores clientes, sin dar opción a realizar acciones cotidianas que presta un computador, esto hasta que se desbloquee los computadores.

Esta pantalla principal es la que se mostrará en el computador administrador la cual contará con las siguientes opciones:

- Administrar
- Elegir Pc
- Pc Aleatoria

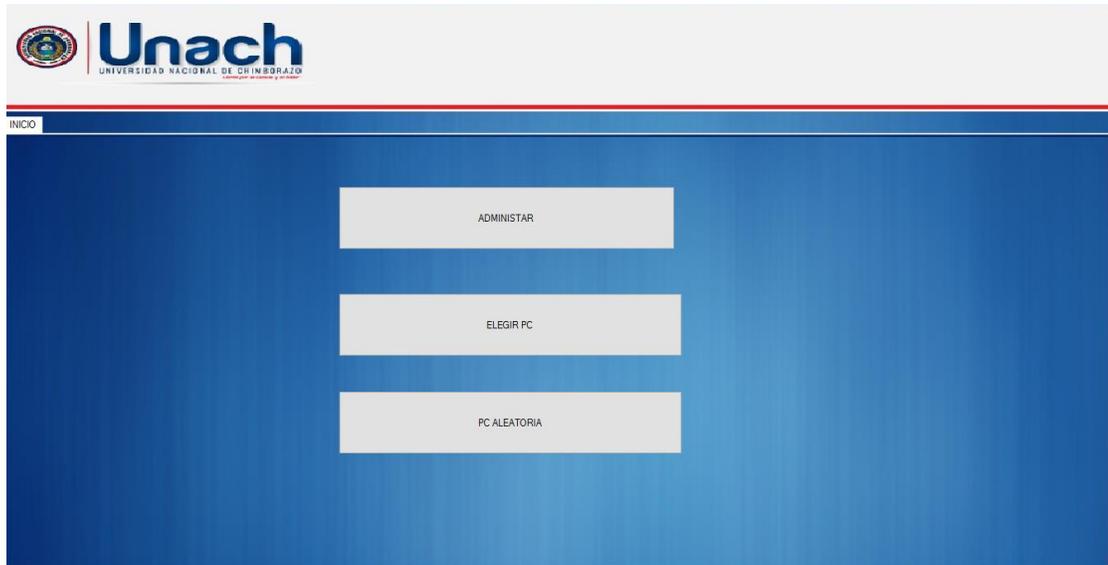


Figura 11–2: Pantalla principal.

Fuente: Velastegui , Lenin, 2016.

Autenticación Cliente y/o servidor

Nos permitirá ingresar al sistema mediante un usuario y una contraseña, abriendo las opciones correspondientes en caso de ingresar como cliente o administrador, posterior a la validación de la información ingresada al sistema.

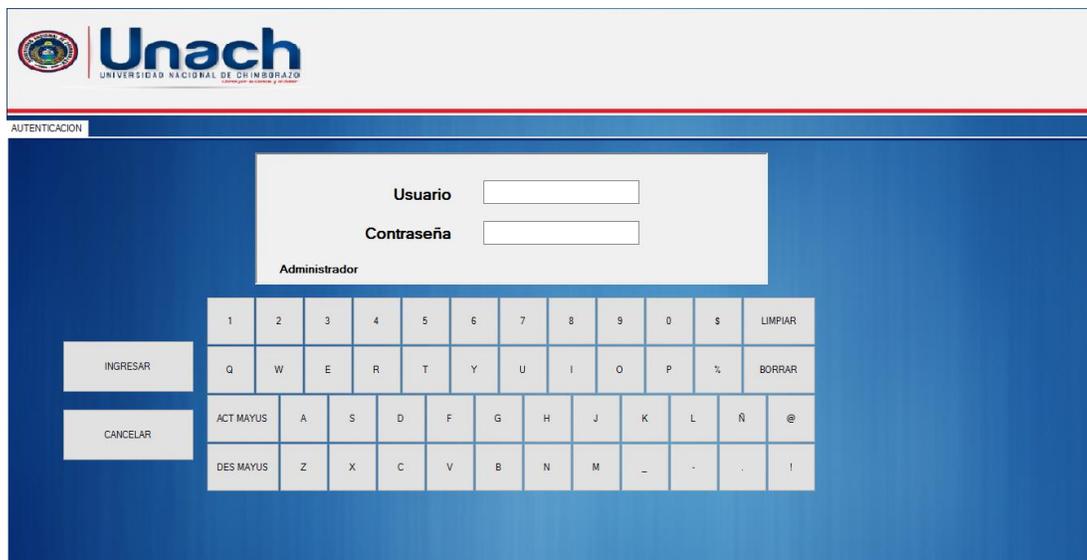


Figura 12–2: Pantalla de login.

Fuente: Velastegui , Lenin, 2016.

Mostrar u ocultar botones clientes en caso se conecten o desconecten

Permitirá visualizar los computadores que estén disponibles dentro del panel que solo los usuarios identificados por el sistema podrán ver, además este panel se actualiza constantemente debido a que existirán computadores clientes que se conecten y otras se desconecten.



Figura 13–2: Panel computadores disponibles.

Fuente: Velastegui , Lenin, 2016.

Módulo de administrador

Acciones Administrador

Luego de la correspondiente autenticación en el sistema, se mostrará el panel con las distintas acciones que puede realizar el administrador. Las acciones son las siguientes.

Tiempo

Uso libre

Reportes

Activar / Desactivar

Reiniciar

Apagar

En la figura 14 – 2: Interfaz de acciones del administrador, podemos observar las acciones separadas por botones.

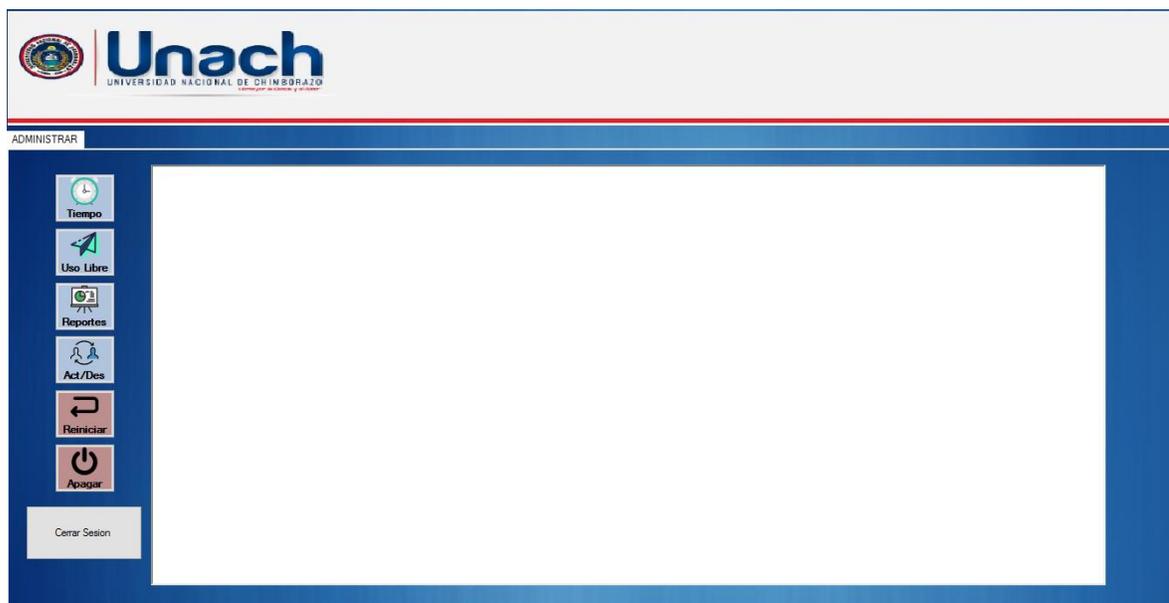


Figura 14–2: Acciones del Administrador

Fuente: Velastegui , Lenin, 2016..

Módulo de cliente

Acciones del cliente

Entre las acciones que puede realizar el cliente están la de desbloquear las pcs dando click sobre el pc que desee desde el panel que muestra las pcs disponibles, desde la opción en la pantalla principal de asignación de una pc en forma aleatoria, y desde la interfaz del app cliente ingresando sus usuario y contraseña.

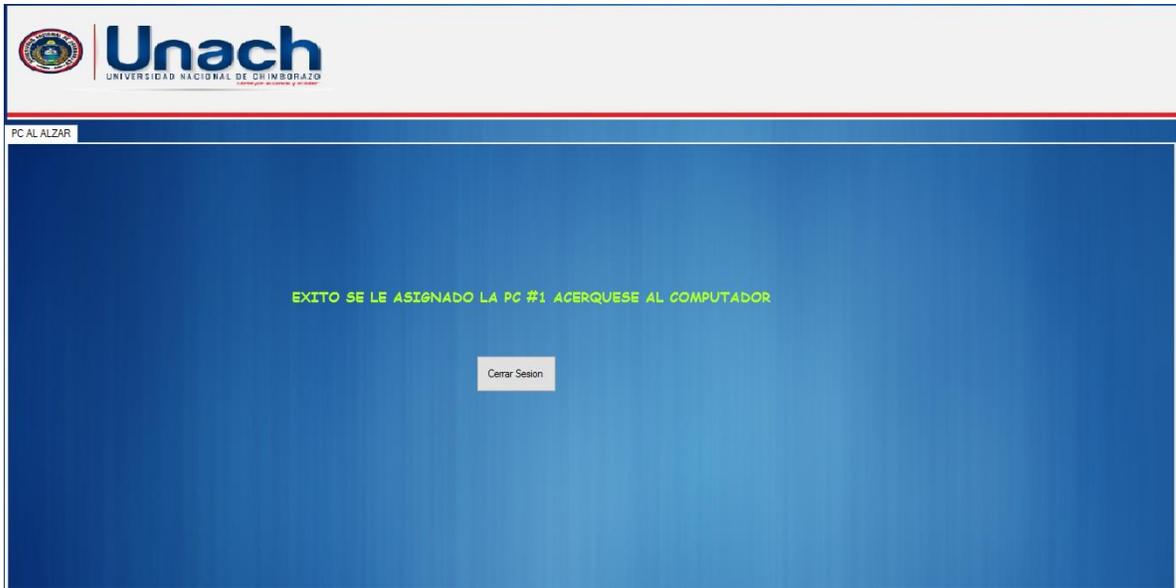


Figura 15–2: Pantalla pc aleatoria

Fuente: Velastegui , Lenin, 2016.

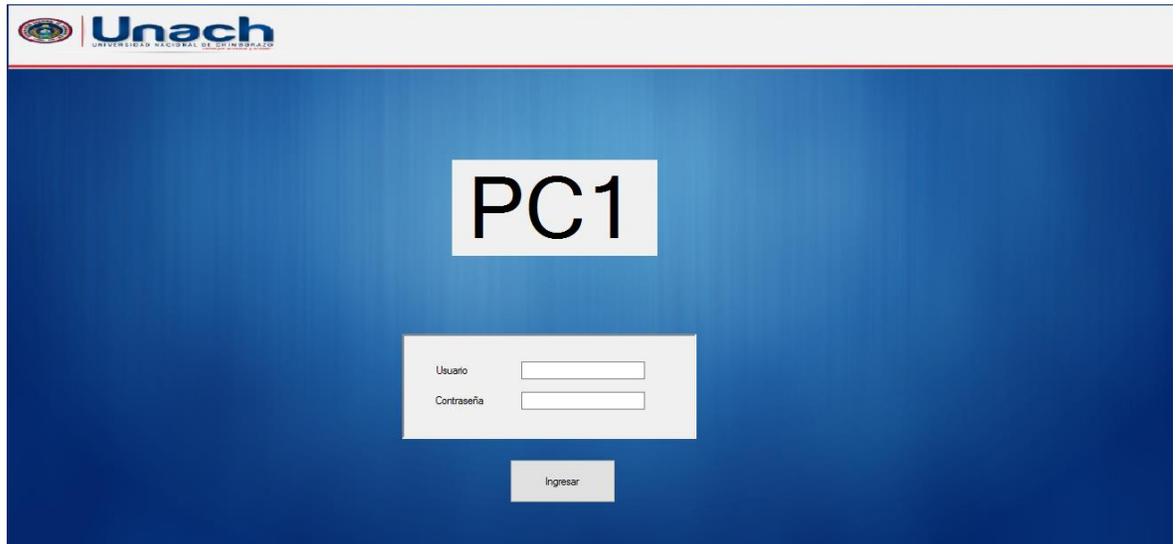


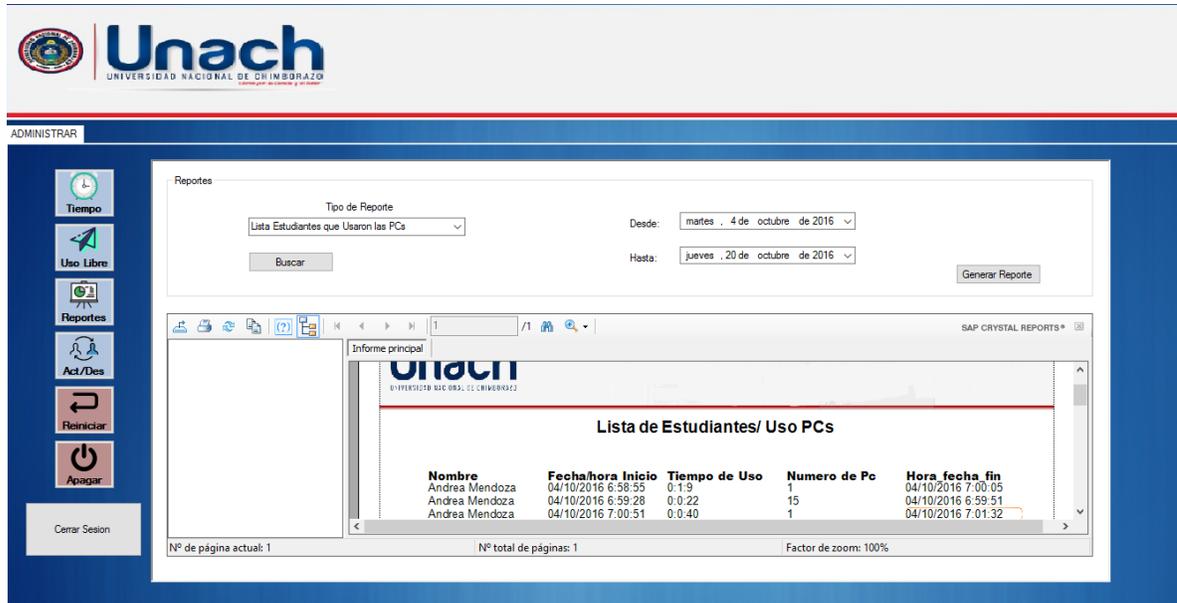
Figura 16–2: Pantalla login del lado cliente

Fuente: Velastegui , Lenin, 2016.

Módulo de reportes

Reportes usuarios que usaron las pcs en un determinado tiempo

El sistema permite generar reportes con una lista obtenida de la base de datos filtrado por medio de dos fechas.



The screenshot displays the 'Reportes' section of the Unach administrative system. The interface includes a sidebar with navigation icons for 'Tiempo', 'Uso Libre', 'Reportes', 'Act/Des', 'Reiniciar', and 'Apagar'. The main content area shows a form to generate a report. The 'Tipo de Reporte' is set to 'Lista Estudiantes que Usaron las PCs'. The date range is from 'martes, 4 de octubre de 2016' to 'jueves, 20 de octubre de 2016'. A 'Generar Reporte' button is visible. Below the form, a preview of the report is shown, titled 'Lista de Estudiantes/ Uso PCs'. The report contains a table with the following data:

Nombre	Fecha/hora Inicio	Tiempo de Uso	Numero de Pc	Hora_fecha_fin
Andrea Mendoza	04/10/2016 6:58:55	0:1:9	1	04/10/2016 7:00:05
Andrea Mendoza	04/10/2016 6:59:28	0:0:22	15	04/10/2016 6:59:51
Andrea Mendoza	04/10/2016 7:00:51	0:0:40	1	04/10/2016 7:01:32

Figura 17–2: Reportes Usuarios por fechas

Fuente: Velastegui, Lenin, 2016.

Reportes de usuarios bloqueados

El sistema permite generar reportes con una lista de usuarios que se encuentren bloqueados en el sistema.

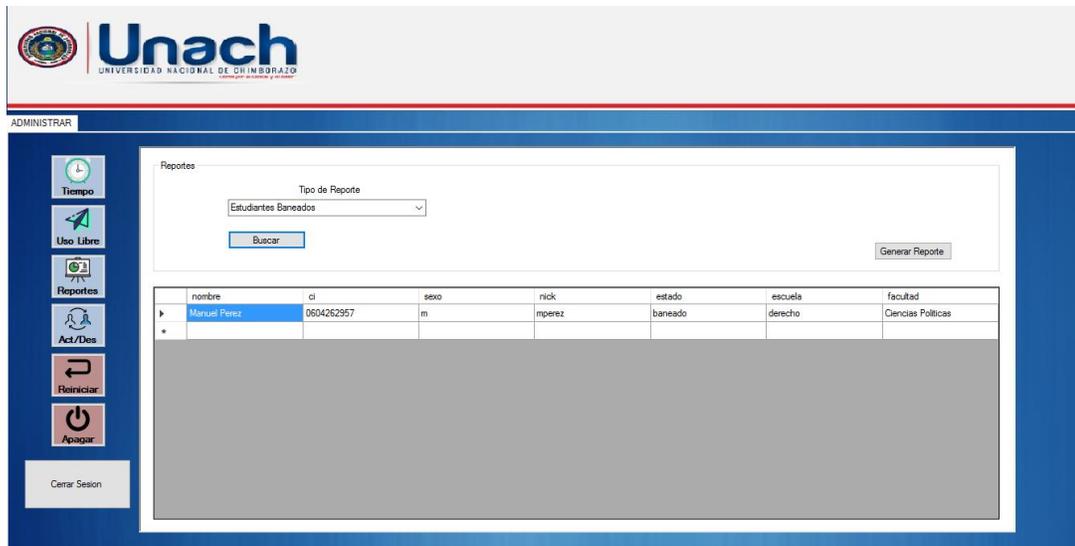


Figura 18–2: Reporte usuarios bloqueados

Fuente: Velastegui , Lenin, 2016.

2.2.8. *Diseño de la base de datos*

En esta etapa se debe tener en cuenta todas las entradas y salidas plasmadas en el documento de análisis de requerimientos de software.

Como podemos observar en la figura 19 – 2, la base de datos consta de 5 colecciones: escuelas, facultades, tipousuarios, usuarios y resgistroso.

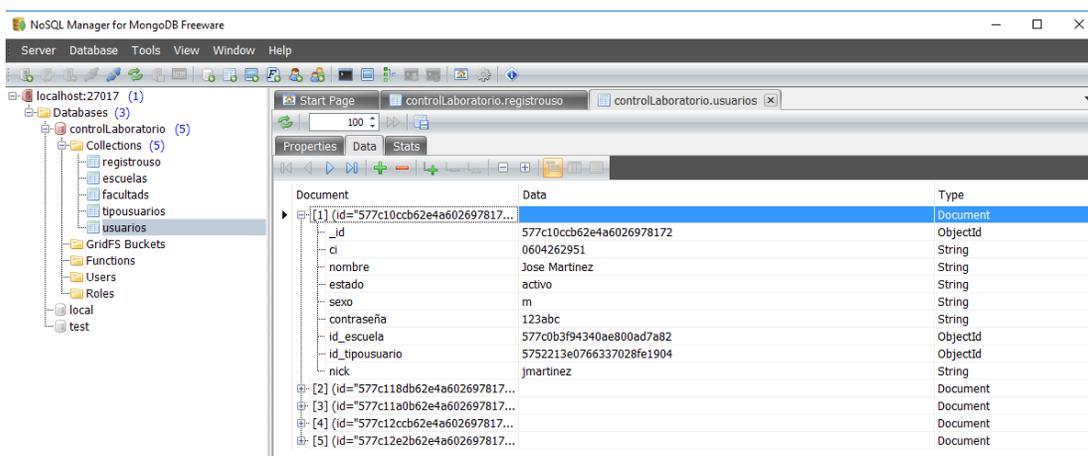


Figura 19–2: Diseño de Base de datos

Fuente: Velastegui , Lenin, 2016.

2.3. Fase de desarrollo.

Durante el proceso de desarrollo la metodología Scrum adopta iteraciones o sprints que contienen diferentes historias de usuario a ser desarrolladas, en el proyecto se realizaron 4 sprints el primero de actividades previas y los siguientes constan las actividades realizadas para crear el sistema.

En total se realizaron 31 historias de usuario que poseen el siguiente formato:

2.3.1. Pila del sprint

En este punto se va asignar cada uno de los requerimientos, identificados en la pila del producto a un determinado sprint, como se puede observar en la siguiente tabla.

Tabla 7 – 2: Pila del sprint

Sprint	Descripción	Fecha Inicio	Fecha Fin
Sprint 0: Acciones Preliminares.	Corresponde a las actividades de planificación del proyecto	30-05-2016	23-06-2016
Sprint 1: Acciones del administrador	Corresponde al manejo de toda la acción que se puede realizar la persona que administra el sistema. Apagado reinicio de todos los computadores. Bloqueo y desbloqueo de los computadores.	24-06-2016	22-07-2016
Sprint 2: Acciones del estudiante	Corresponde al manejo de toda la acción que se puede realizar los estudiantes en el sistema. . Bloqueo y desbloqueo de los computadores.	25-07-2016	19-08-2016
Sprint 3: Administración de control y Reportes	Administración de los datos concernientes a los usuarios que manipularan el sistema. Generación de los reportes.	22-08-2016	19-09-2016

Fuente: Velastegui , Lenin, 2016.

En la siguiente tabla se puede observar las historias de usuario correspondiente a las acciones del administrador los cuales están agrupados en el sprint 1, las demás historias de usuarios se encontrarán en el ANEXO B.

Tabla 8 – 2: Sprint 1 – Acciones del administrador

Sprint 1 – Acciones del Administrador	
ID	Descripción
1	Como administrador deseo poder ingresar al sistema a través de un username y contraseña.
2	Como administrador deseo digitar mi usuario y contraseña de forma táctil.
3	Como administrador y/o estudiante deseo tener una pantalla principal donde se muestren las opciones de ingreso al sistema.
4	Como administrador deseo tener una interfaz donde se muestren las opciones que pueda realizar.
5	Como administrador deseo poder desbloquear todos los computadores por un determinado tiempo ingresado de antemano.
6	Como administrador deseo poder desbloquear todos los computadores hasta cuando elija bloquearlos.
7	Como Administrador deseo obtener una lista de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).
8	Como Administrador deseo obtener una lista de los estudiantes que no pueden ingresar al sistema.
9	Como administrador deseo poder bloquear o desbloquear el ingreso al sistema.
10	Como administrador deseo reiniciar todos los computadores que estén conectados a la maquina principal.
11	Como administrador deseo apagar todos los computadores que estén conectados a la maquina principal.
12	Como administrador deseo cerrar sesión en el sistema y volver a la pantalla de inicio.

Fuente: Velastegui , Lenin, 2016.

Tabla 9 – 2: Historia de Usuario HU01-S1

HU01-S1: Como administrador deseo poder ingresar al sistema a través de un username y contraseña.			
Descripción	Se requiere realizar una autenticación con el usuario y contraseña que proporcione el administrador, para que pueda ingresar al sistema.		
Valor del negocio	10		
Puntos Estimados	15	Puntos Reales	10
Criterio de Aceptación	Datos ingresados incorrectos, alerta de error en los datos. Datos correctos accede a la pantalla de selección de acciones administrativas. Verificación de campos vacíos. Mensaje de bienvenida.		
Tareas	T1-HU01-S1: Creación de la base de datos. T2-HU01-S1: Creación de la colección usuarios. T3-HU01-S1: Creación de manager para usuarios. T4-HU01-S1: Creación de la interfaz principal con la opción administrar. T5-HU01-S1: Creación de métodos autenticar. T6-HU01-S1: Diseño interfaz de autenticar. T7-HU01-S1: Validaciones. T8-HU01-S1: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

En el casillero superior el texto escrito por el usuario del requerimiento acompañado de un código, en la descripción consta un detalle de lo que se desea que haga el sistema, en los puntos de valor del negocio se plasma en una escala de 1 a 10 cuanto puede afectar para el desempeño esta tarea, en los puntos estimados son valores en horas 1 punto de estimación es 1 hora de trabajo en la tarea designada, estos puntos se estimaron en base a experiencias previas en el desarrollo por parte del Sprint Planning teniendo en cuenta los tiempos de entrega del sistema, los puntos reales plasman el tiempo que en realidad se demoró en realizar la tarea lo ideal es que los puntos reales sean lo más similares a los estimados, en el criterio de aceptación consta la forma de responder el sistema frente a esos eventos y al final se subdividen tareas para que el equipo de desarrollo pueda realizar, a continuación las demás historias de usuario.

Las historias de usuario restantes las podemos encontrar en el ANEXO C.

2.3.2. *Tablero de tareas*

Dentro de la metodología SCRUM se contempla la creación de un tablero en el cual se puede visualizar la totalidad de las tareas desprendidas de cada una de las HU, además la idea principal es comunicar el progreso de cada una de ellas. De esta forma el equipo tiene conocimiento del desempeño de cada uno de los miembros del proyecto, lo cual permite marcar el ritmo de trabajo del equipo. En la siguiente tabla está representado el Tablero de tareas, donde podemos observar el progreso de cada historia de usuario.

Tabla 10 – 2: Tablero de tareas

Pila del producto	Por hacer	En progreso	Terminadas
HU01-S1: Como administrador deseo poder ingresar al sistema.			T1-HU01-S1: Creación de la base de datos. Lenin Velastegui.
			T2-HU01-S1: Creación de la colección usuarios. Lenin Velastegui.
			T3-HU01-S1: Creación de manager para usuarios. Lenin Velastegui.
			T4-HU01-S1: Creación de la interfaz principal con la opción administrar. Lenin Velastegui.
			T5-HU01-S1: Creación de métodos autenticar. Lenin Velastegui.
			T6-HU01-S1: Diseño interfaz de autenticar. Lenin Velastegui.
			T7-HU01-S1: Validaciones. Lenin Velastegui.
			T8-HU01-S1: Pruebas. Lenin Velastegui.

Fuente: Velastegui , Lenin, 2016.

2.3.3. Pruebas de aceptación

Luego de que ya se ha construido el sistema lo que resta determinar si el resultado obtenido, establecido previamente en las HU (criterios de aceptación) cumple con lo planificado.

En la tabla 11 – 1, se puede apreciar el formato utilizado para realizar las pruebas de aceptación de cada una de las HU.

Tabla 11– 2: Prueba de ingreso al sistema administrador.

Ficha de prueba	
Prueba – 01: Ingreso al sistema.	
Código de HU: HU01-S1	
Descripción de HU: Como administrador deseo ingresar al sistema.	
Estado: Superado	
Evento	Valor Esperado
Autenticación ingresando datos al sistema.	Ingreso al sistema, mostrando las acciones que puede realizar el administrador.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

En esta sección se encuentra el número de referencia de la prueba, en la descripción va el nombre del requerimiento a probar, las condiciones vienen a ser los criterios de aceptación para realizar las pruebas y lo más importante de documentar sería los descriptivos del resultado esperado y el resultado obtenido durante las pruebas, con ello si la prueba no fuese exitosa se puede rehacer la tarea o simplemente se toma la prueba como aceptada, en total se realizaron pruebas una por cada historia de usuario, que se muestran a continuación.

El restante de las pruebas de aceptación se lo puede encontrar en el ANEXO E.

2.4. Implantación.

Durante el proceso de implantación se utilizó para instalar la aplicación administradora una computadora hp con procesador Intel core i7 de 3.2 Ghz de 6º generación, con 8 Gb de memoria ram DDR3 y disco duro de 1Tb con pantalla táctil, para los clientes se utilizaron las computadoras que se encuentran en la sala de internet del CTE y para el servidor una computadora Hp Intel core i7 la cual va a contener el servidor de Node.js y socket.io, con todos estos equipos se procedió a montar el ambiente instalando la aplicación administrador, cliente y sus respectivas librerías, una vez establecido se procede a desplegar la solución y se realizó una prueba final de comunicación entre base de datos y las aplicaciones.

CAPITULO III

3. MARCO DE RESULTADOS

3.1. Gestión del proyecto

Scrum considera fundamental el monitoreo del avance del proyecto, el tener la posibilidad de apreciar la velocidad de desarrollo del proyecto, permitiendo analizar la eficiencia en el desarrollo del mismo. Para poder evaluar el proyecto se utiliza una herramienta conocida como Burndown Chart, el mismo es un gráfico que contrasta la planificación en horas con las horas reales que fueron necesarias para cumplir con las tareas. En la figura 1 – 3 podemos observar cómo se llevó a cabo el desarrollo del proyecto.

Tabla 1 - 3: Horas de trabajo por semana

Actividad	TIEMPO ESTIMAD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	TIEMPO REAL TOTAL
1	16	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
2	16	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
3	16	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
4	24	10	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32
5	24	0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32
6	16	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
7	16	0	0	24	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34
8	8	0	0	10	6	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26
9	8	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
10	16	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
11	24	0	0	0	10	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48
12	16	0	0	0	0	10	14	0	15	0	0	0	0	0	0	0	0	0	0	0	0	39
13	8	0	0	0	0	0	8	0	0	15	0	0	0	0	0	0	0	0	0	0	0	23
14	16	0	0	0	0	0	22	10	0	0	0	0	0	0	0	0	0	0	0	0	0	32
15	16	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	30
16	24	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0	0	0	0	0	32
17	32	0	0	0	0	0	0	0	20	40	4	0	0	0	0	0	0	0	0	0	0	64
18	32	0	0	0	0	0	0	0	0	5	30	25	0	0	0	0	0	0	0	0	0	60
19	16	0	0	0	0	0	0	0	0	0	10	10	10	0	0	0	0	0	0	0	0	30
20	32	0	0	0	0	0	0	0	0	0	0	10	14	0	0	0	0	0	0	0	0	24
21	48	0	0	0	0	0	0	0	0	0	0	0	20	22	30	0	0	0	0	0	0	72
22	32	0	0	0	0	0	0	0	0	0	0	0	0	24	10	10	20	10	0	0	0	74
23	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	20	0	0	0	0	48
24	48	0	0	0	0	0	0	0	0	0	0	0	0	20	0	12	5	23	0	0	0	60
25	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	30	10	0	0	70
26	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	12	0	32
27	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	12	42
28	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	30
29	24	0	0	0	10	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	4	24
30	32	0	0	0	0	0	0	0	10	0	0	10	0	0	0	0	0	0	0	0	4	24
31	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	0	10	0	2	32
	778	44	54	50	60	58	44	50	77	60	44	55	44	66	40	60	55	63	60	52	52	1088

Fuente: Velastegui, Lenin, 2016.

A continuación, se muestran los valores resultantes tomando en cuenta los valores estimados y los valores reales.

Tabla 2 - 3: Tiempos reales y estimados por semana

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	
HORAS ESTIMAC	1088	1044	990	940	880	822	778	728	651	591	547	492	448	382	342	282	227	164	104	52	0 suma
HORAS REALES	778	739,1	700,2	661,3	622,4	583,5	544,6	505,7	466,8	427,9	389,0	350,1	311,2	272,3	233,4	194,5	155,6	116,7	77,8	38,9	0,0 ideal

Fuente: Velastegui , Lenin, 2016.

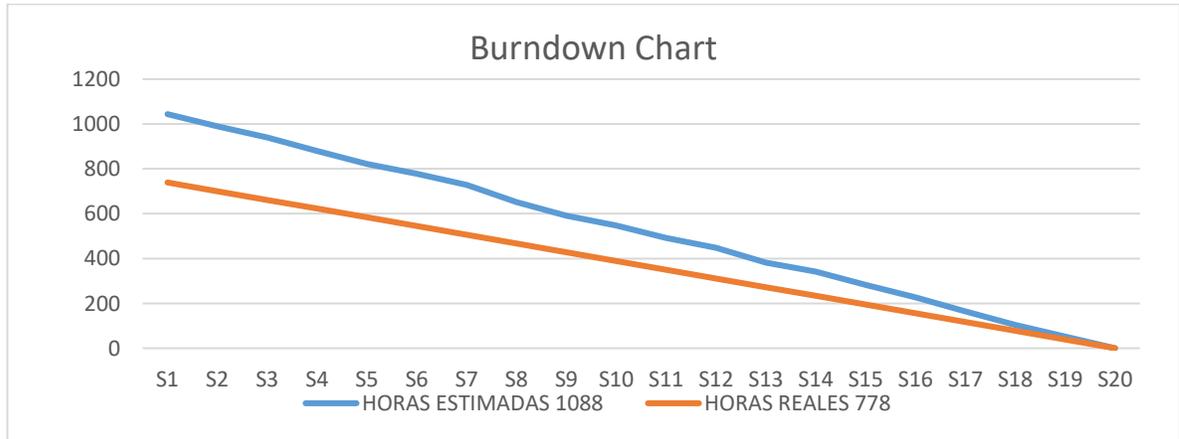


Figura 1 – 3: Burndown Chart

Fuente: Velastegui , Lenin, 2016.

Se nota la compensación de la parte superior de un trabajo mayor de horas con respecto a los problemas tenidos al final en las implementaciones y pruebas con lo que se puede observar una buena gestión del riesgo en cuanto a estimaciones de tiempo y esfuerzo que se requiere para terminar el proyecto en las semanas planificadas.

3.2. Aplicación del Estándar ISO 9126 para la evaluación del sistema de control de accesos.

La ISO 9126 es un estándar creada para la especificación y evolución extensiva de la calidad de los productos de software especificando características relevantes de calidad. Este estándar proporciona un marco de trabajo para evaluar la calidad de todo tipo de productos de software apoyándose en las siguientes métricas de evaluación.

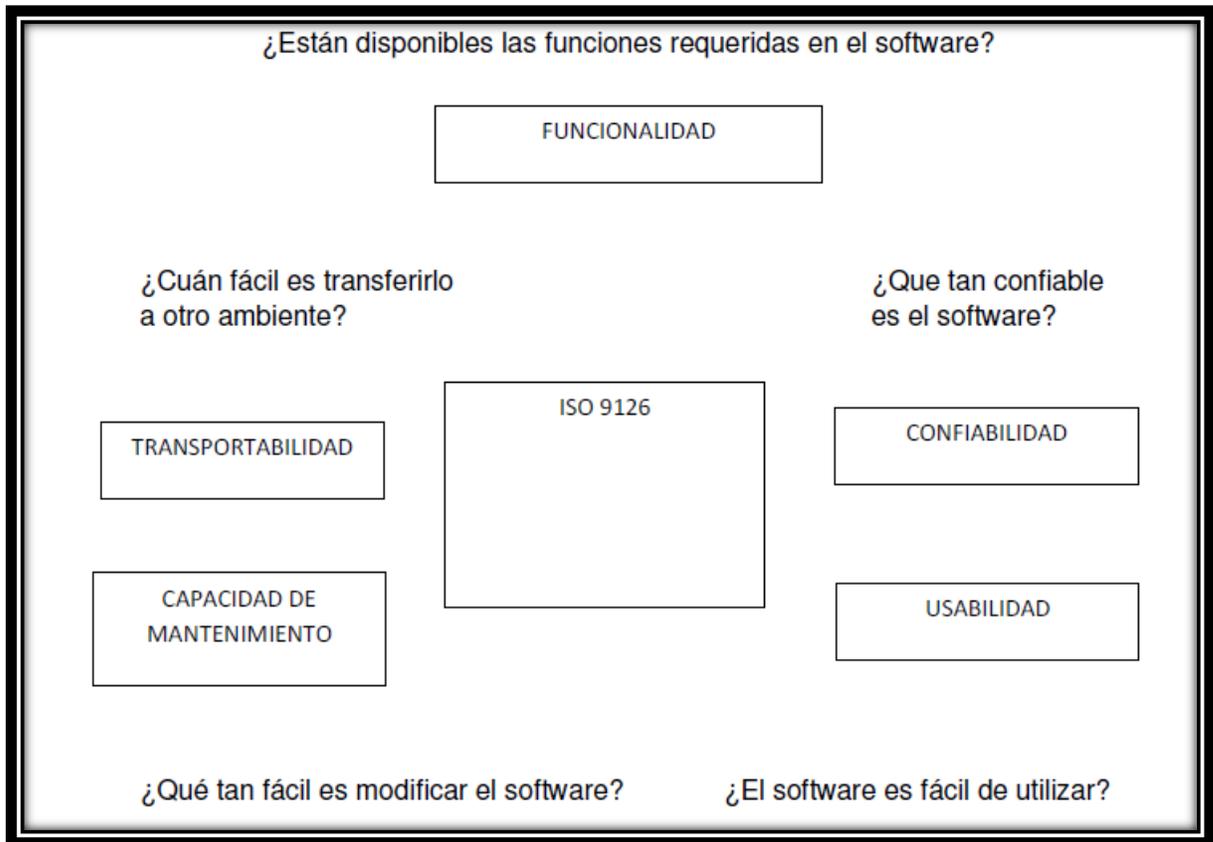


Figura 2 – 3: Indicadores del estándar ISO 9126

Fuente: Velastegui , Lenin, 2016.

Para la evaluación de cada una de las métricas y sus características se toma valores de cero ausentes y uno que el sistema cumple con la característica.

Para la evaluación de cada una de las métricas del estándar ISO 9126 se necesitó de la ayuda de la persona que va a estar encargada del sistema (Administrador) y de un grupo de estudiantes los cuales pertenecen al tercer semestre paralelo “A” de especialidad de ingeniería en sistemas de la UNACH. - El grupo de estudiantes está conformado por 38 integrantes y evaluarán las métricas correspondientes a los usuarios normales.

Las métricas correspondientes a cada uno de los evaluados serán de la siguiente manera:

- Funcionalidad – Administrador.

- Confiabilidad – Administrador.
- Utilidad – Grupo de estudiantes.
- Eficiencia – Administrador.
- Mantenibilidad – Administrador.
- Portabilidad – Administrador.

Funcionalidad

Tabla 3 - 3: Funcionalidad

FUNCIONALIDAD			
Características	Preguntas	Cumplimientos	Observaciones
Conveniencia	¿El software desempeño las tareas requeridas?	1	El software cumplió con el total de requerimientos del usuario.
Precisión	¿El resultado es el esperado?	1	
Interoperabilidad	¿El sistema puede Interactuar con otro?	0	
Seguridad	¿El sistema impide el acceso a personas no autorizadas?	1 1	El sistema no permite la utilización de los computadores, si estas no están registradas en la base de datos del sistema
Total		4/5	80%

Fuente: Velastegui , Lenin, 2016.

Confiabilidad

Tabla 4 - 3: Confiabilidad

CONFIABILIDAD			
Características	Preguntas	Cumplimientos	Observaciones
Vencimiento	¿Muchas de las fallas han sido eliminadas durante el tiempo?	1	Cuando se encontraba un error se lo corregía inmediatamente
Tolerancia a las fallas	¿El software es capaz de manejar de errores?	1	El sistema notifica de los errores que se presenten, contiene try and catch
Capacidad de recuperación	¿Puede el software reasumir el funcionamiento y restaurar datos perdidos después de la falla?	0	
	Total	2/3	66%

Fuente: Velastegui , Lenin, 2016.

Utilidad

Tabla 5 - 3: Utilidad

UTILIDAD					
Características	Preguntas	Si	No	% Afirmativas	% Negativas
Claridad	¿Puede usted utilizar fácilmente el sistema?	36	2	94.7	5.3
Capacidad de aprendizaje	¿Piensa usted que sea fácil aprender a usar el sistema?	38	0	100	0
Operatividad	¿Puede usted utilizar el sistema sin mucho esfuerzo?	34	4	89.5	10.5
Atractivo	¿La interfaz se ve bien?	32	6	84.2	15.8
	Total			92.1 %	7.9 %

Fuente: Velastegui , Lenin, 2016.

Eficiencia

Tabla 6 - 3: Eficiencia

EFICIENCIA			
Características	Preguntas	Cumplimientos	Observaciones
Comportamiento del tiempo	¿Qué tan rápido responde el sistema?	1	
Utilización de recursos	¿El sistema utiliza los recursos de manera eficiente?	1	No consume mucha memoria tampoco utiliza mucho procesamiento.
	Total	2/2	100%

Fuente: Velastegui , Lenin, 2016.

Para la utilización de recursos se revisó cuanto utiliza el sistema en memoria RAM y CPU en tiempos en los que las aplicaciones administrador y cliente se encuentran en los estados de reposo y al momento de utilizarlas, dando los siguientes resultados.

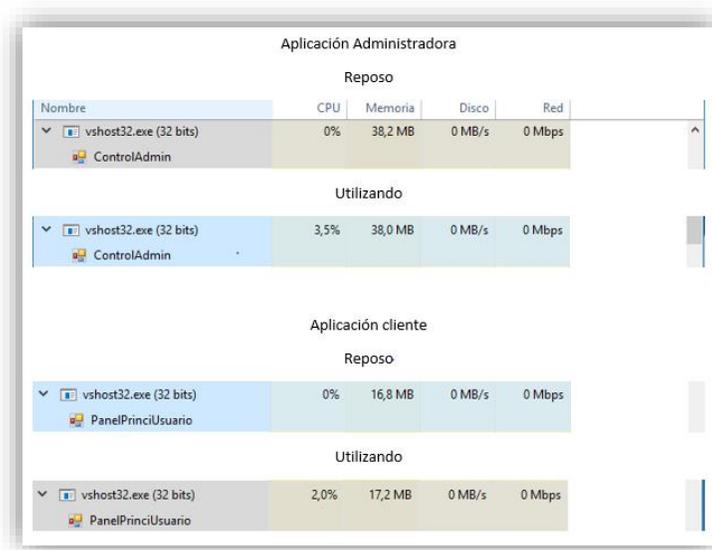


Figura 3 – 3: Utilización de recursos por parte del sistema

Fuente: Velastegui , Lenin, 2016.

En la figura 2-3 podemos observar que el sistema no utiliza muchos recursos, ocupan un rango de 0 a 3.5% de procesamiento y entre 16.8 a 38.2 de memoria RAM de los computadores donde se encuentran implementados.

Mantenibilidad

Es la capacidad del producto del software para ser modificado.

Tabla 7 - 3: Mantenibilidad

CAPACIDAD DE MANTENIMIENTO			
Características	Preguntas	Cumplimientos	Observaciones
Capacidad de análisis	¿Las fallas pueden ser fácilmente diagnosticadas?	0	En caso de falla se deben revisar lo que muestra la consola de servidor para eso toca entrar al cmd
Variabilidad	¿El sistema puede ser fácilmente modificado?	1	Si el sistema está desarrollado con herramientas que permiten flexibilidad como Node.js para poder instalar paquetes y mongodb modificar la estructura de la base de datos
Estabilidad	¿El sistema puede seguir funcionando si se hacen cambios?	1	
Capacidad de prueba	¿El sistema puede ser probado fácilmente?	1	Se puede probar ejecutando directamente en visual Studio.
	Total	3/4	75%

Fuente: Velastegui , Lenin, 2016.

Portabilidad

Capacidad del software para ser trasladado de un entorno a otro.

Tabla 8 - 3: Portabilidad

PORTABILIDAD			
Características	Preguntas	Cumplimientos	Observaciones
Adaptabilidad	¿El software se puede trasladar a otros ambientes?	0	
Capacidad de instalación	¿El software se puede instalar fácilmente?	1	
Conformidad	¿El sistema puede seguir funcionando si se hacen cambios?	1	
Capacidad para reemplazar	¿El software puede reemplazar fácilmente otro software?	1	Siempre y cuando sea para el control de acceso a computadores
	Total	3/4	75%

Fuente: Velastegui , Lenin, 2016.

Para obtener el porcentaje total de la evaluación aplicando el estándar ISO 9126 se utilizó la siguiente operación $((\%afirmativoM1 + \%afirmativoM2 + \%afirmativoM3 + \%afirmativoM4 + \%afirmativoM5 + \%afirmativoM6) / 6)$ dando como resultados los siguientes valores.



Figura 4 – 3: Análisis Estándar ISO 9126

Fuente: Velastegui , Lenin, 2016.

Análisis: Tomando en cuenta los factores positivos y negativos identificados en las encuestas se obtuvieron el 81% de factores positivos en cuanto usabilidad y el 19 % de factores negativo por lo que el sistema cumple en una mayor parte del estándar de calidad ISO 9126.

3.3. Validación Usabilidad

La evaluación de la usabilidad del sistema se la realizó mediante un test en el cual se evaluó identidad, contenido, navegación, utilidad y retroalimentación.

Tabla 9 - 3: Identidad

Identidad	Si	No
¿Se plasma en la pantalla de inicio la identidad de la Universidad Nacional de Chimborazo expresada en colores?	1	0
¿Hay algún elemento gráfico o de texto que le haya ayudado a entender más claramente a qué universidad pertenece?	1	0
¿El sistema cumple con todos los requerimientos exigidos por el CTE?	1	0
¿De los elementos que muestra esta pantalla, hay algo que usted crea que está fuera de contexto?	1	0
¿En cada una de las pantallas se observa el nombre de la UNACH?	1	0
TOTAL:	5	0

Fuente: Velastegui , Lenin, 2016.

Análisis: Se capacitó al personal del CTE que utilizará el sistema y en base a lo respondido se obtiene que 5 de 5 preguntas que posee el sistema fueron identificadas de forma fácil, lo que equivale a un 100 %, y evidencia que cumple el sistema pertenece a la UNACH.

Tabla 10 - 3: Contenido

Contenido	Si	No
¿La aplicación cumple con los contenidos previstos?	1	0
¿Al ver el sistema, pudo distinguir de una sola mirada cuál era el contenido más relevante que se ofrecía?	1	0
¿Se identifica de forma clara el ingreso de datos correcto?	1	0
¿Los datos a ser ingresados son suficientemente descriptivos?	1	0
¿No se repite información en el sistema?	1	0
TOTAL:	5	0

Fuente: Velastegui , Lenin, 2016.

Análisis: En la tabla se tiene un total de 5 respuestas positivas lo que equivale al 100% del cumplimiento del contenido previsto fácil de entender y descriptivo para el usuario, se plasma que el sistema se realizó de forma personalizada y en conjunto con el usuario.

Tabla 11 - 3: Utilidad

Utilidad	Si	No
¿Al observar la pantalla de inicio se puede avizorar de lo que trata el sistema? y ¿Qué ofrece?	1	0
¿Cree que los contenidos y servicios que se ofrecen en este sistema son de utilidad para su caso personal?	1	0
TOTAL:	2	0

Fuente: Velastegui , Lenin, 2016.

Análisis: La utilidad del sistema se basa en la solución de problemas que se plante solucionar durante el proceso de facturación por ello tiene una valoración del 100% de cumplimiento.

Tabla 12 - 3: Navegación

Navegación	Si	No
¿La forma en que se navega por el sistema y sus opciones, es fácil de realizar?	1	0
¿Existen opciones a parte del login en la aplicación principal para realizar la autenticación de estudiante?	1	0
¿Logra distinguir gráficamente los datos que ya han sido registrados?	0	1
TOTAL:	2	1

Fuente: Velastegui , Lenin, 2016.

Análisis: En cuanto a navegación se muestra un 66.66% lo que es bastante bueno para cumplir dicho criterio permitiendo al usuario poder desempeñar las funciones con el software de forma satisfactoria.

Tabla 13 - 3: Retroalimentación

Retroalimentación	Si	No
¿La aplicación cumple con sus expectativas?	1	0
¿Al momento de registrar los datos en el sistema ¿Ocurrió algún error?	0	1
¿Al momento de generar un reporte en el sistema ¿Ocurrió algún error?	0	1
TOTAL:	1	2

Fuente: Velastegui , Lenin, 2016.

Análisis: Con las respuestas de estas preguntas se nota que cumple con el 100% de aceptación el sistema en cuanto a retroalimentación.



Figura 5 – 3: Análisis de usabilidad

Fuente: Velastegui , Lenin, 2016.

Análisis: Tomando en cuenta los factores positivos y negativos identificados en las encuestas se obtuvieron 17/18 factores positivos en cuanto usabilidad y solo un 1 factor negativo por lo que el sistema cumple con los parámetros básicos de usabilidad para el usuario.

CONCLUSIONES

- Al terminar el proyecto, las actividades que se realizaban de forma manual ahora gracias al sistema de control de accesos se lo realiza en una forma automatizada dando al usuario una nueva experiencia al llevar a cabo el proceso que venía realizando.
- El software ha permitido la reducción del consumo de recursos debido a que se ha optimizado el tiempo al momento de controlar el acceso de los estudiantes a la sala de internet del Centro de Tecnologías Educativas.
- Hoy en día siguen apareciendo nuevas tendencias de programación, es así que para el desarrollo del sistema se utilizó tecnología no convencional, las cuales después de la correspondiente investigación se aplicó en el desarrollo del sistema resultando muy útiles y fáciles de aplicar.
- La metodología Scrum es utilizable para todo tipo de proyectos, esta metodología permitió gestionar cada etapa del desarrollo del sistema, para ello se definieron tres sprints, en la mayoría de los cuales se cumplió con el tiempo establecido y la minoría hubo pequeños retrasos debido a los cambios realizados durante el proceso de desarrollo.

RECOMENDACIONES

- Analizar los requerimientos proporcionados por el usuario, esto ayuda a tener una mejor visión del propósito del proyecto, además permite llevar una adecuada gestión de las tareas a realizar durante el desarrollo del proyecto.
- Se recomienda implementar este tipo de sistemas de control de acceso, ya que en este caso permitirá el control y seguridad de los laboratorios de la Universidad Nacional de Chimborazo, así también ayudará eficientemente en el acceso a los mismos, sin necesidad de la intervención de personas no autorizadas.
- Se sugiere, mejorar este proyecto acoplando un lector de carnets de estudiantes, esto permitirá que la autenticación se la realice de una manera más rápida y eficiente.
- Utilizar la metodología de desarrollo de software SCRUM, ya que permite priorizar los requerimientos del usuario, además que permite observar los resultados al finalizar cada uno de los sprints.
- Antes de implementar un sistema se recomienda, una adecuada capacitación al usuario final del mismo, puesto que de esto dependerá el éxito o fracaso del proyecto.
- Se recomienda realizar la utilización de versionadores de código como por ejemplo GitHub, mismo que permiten además de un trabajo colaborativo, y a la vez un adecuado manejo de las versiones de un proyecto.

BIBLIOGRAFÍA

ARTURO MUÑOS DE LA TORRE. “ *Introduccion a Node.js V8.0*”, [linea], Copyright (C) 2013 Arturo Muñoz de la Torre, pp 7-37.

[Consulta: 05 de julio del 2016]

https://dl.dropboxusercontent.com/u/2990198/nodejskoans/Introduccion_a_Nodejs_a_traves_de_Koans_ebook.pdf

ARTURO MUÑOS DE LA TORRE. “ *Introduccion a Socket.io*”, [linea], Copyright (C) 2013 Arturo Muñoz de la Torre, pp 157-188.

[Consulta: 07 de Agosto del 2016]

https://dl.dropboxusercontent.com/u/2990198/nodejskoans/Introduccion_a_Nodejs_a_traves_de_Koans_ebook.pdf

BRETZ, A., & IHRIG, C. “*FULL STACK JAVASCRIPT DEVELOPMENT WITH MEAN . UNITED STATES OF AMERICA.*” . SITEPOINT., 2015.

[Consulta: 23 de julio del 2016]

COMMUNITY EXPERIENCE DISTILLED. “Socket.IO Real-time web application development”, [linea], Birmingham-Mumbai, 2013, pp 26-42.

[Consulta: 07 de Agosto del 2016]

<http://www.doc-developpement-durable.org/file/Projets-informatiques/cours-&-manuels-informatiques/Web/Socket.IO%20Real-time%20Web%20Application%20Development.pdf>.

SEBATIEN PUTIER. “C# 6 y Visual Studio 2015”, [linea], diciembre 2015.

[Consulta: 20 Agosto del 2016]

<http://www.ediciones-eni.com/libro/c-6-y-visual-studio-2015-los-fundamentos-del-lenguaje-9782746099364>

DOINTECH. “*Sistemas de Control de Accesos*”. [linea]., 2016.

[Consulta: 09 de Agosto del 2016]

<http://www.dointech.com.co/control-de-acceso.html>

DE LA TORRE, C., & GONZALEZ, R. “*ARQUITECTURA SOA CON TEGNOLOGÍA MICROSOFT.*” ESPAÑA: KRASIS PRESS, 2005.

[Consulta: 16 de julio del 2016]

FUNCION 13. “*Ya puedes empezar a aprender Node.js... ¡en castellano!*”. [linea]., 2016.

[Consulta: 21 de Agosto del 2016]

<https://www.funcion13.com/ya-puedes-empezar-a-aprender-node-js-en-castellano/>

HERNÁNDEZ SAMPIERI, R. “*Medotodologia de la investigacion.*” [linea]., 2011.

[Consulta: 21 de julio del 2016]

<http://www.dgsc.go.cr/dgsc/documentos/cecadades/metodologia-de-la-investigacion.pdf>

KIMALDI. “*Control de asistencia a centros educativos mediante terminales de reconocimiento facial*” [linea]., 2016.

[Consulta: 19 de Agosto del 2016]

http://www.kimaldi.com/sectores/universidades/control_de_asistencia_a_centros_educativos_mediante_terminales_de_reconocimiento_facial

LINKED. “*PARA QUE SIRVE UN SISTEMA DE CONTROL DE ACCESO?*”. [linea], 2016.

[Consulta: 15 de Agosto del 2016]

<https://www.linkedin.com/pulse/para-que-sirve-un-sistema-de-control-acceso-rovillel-sandoval>

MONGODB, INC. “*MongoDB.*” .[linea], 2016.

[Consulta: 21 de julio del 2016]

<https://www.mongodb.org/>

MONGODB. “*The MongoDB Documentation Project no longer publishes documentation in the PDF format.*” .[linea], 2016.

[Consulta: 09 de Agosto del 2016]

<https://docs.mongodb.com/manual/meta/pdfs/>

NEWTONSOFT. “*Newtonsoft JSON.NET.*” .[linea], 2016.

[Consulta: 25 de julio del 2016]

<http://www.newtonsoft.com/json>

NODE.JS FOUNDATION. “*About Node.js.*” .[linea], 2016.

[Consulta: 29 de julio del 2016]

<https://nodejs.org/>

NODE.JS HISPANO. “*Introducción a Socket.io #nodejs.*” .[linea], 2016.

[Consulta: 28 de Agosto del 2016]

<http://www.nodehispano.com/2012/09/introduccion-a-socket-io-nodejs/>

ORNBO, GEORGE. “¿Programacion Node.js?”. Madrid : GRUPO ANAYA,S.A., 2013, pp 25-29.

[Consulta: 10 julio del 2016]

ORNBO, GEORGE. “¿Programacion Node.js?”. GRUPO ANAYA,S.A., 2013, pp 33-45.

[Consulta: 15 julio del 2016]

ORNBO, GEORGE. “¿Programacion Node.js?”. Madrid : GRUPO ANAYA,S.A., 2013, pp 213-256.

[Consulta: 27 julio del 2016]

ROGER, P. “*SOFTWARE ENGIENEERING.*”. Boston: McGraw-Hill., 2001.

[Consulta: 05 de julio del 2016]

ROSELL TEJADA, J. M., & VILLALÓN HUERTA., A. “*shutdown.*”. [linea]., 2015.

[Consulta: 23 de julio del 2016]

www.shutdown.es/une195.pdf

SUDARSHAN, S., KORTH, H., & SILBERSCHATZ, A. “*FUNDAMENTOS DE BASES DE DATOS.*”. Madrid: McGRAW-HILL., 2002.

[Consulta: 15 de julio del 2016]

SOCKET.IO. “*socket.io.*”. [linea]., 2016.

[Consulta: 03 de julio del 2016]

<http://socket.io/>

WIKIPEDIA.ORG. “Microsoft Visual Studio”[línea], 27 de 06 de 2016.

[Consulta: 10 julio del 2016]

https://es.wikipedia.org/wiki/Microsoft_Visual_Studio.

WIKIPEDIA.ORG. “Interfaz táctil de usuario.”[línea], 24 de 04 de 2015.

[Consulta: 13 julio del 2016]

https://es.wikipedia.org/wiki/Interfaz_t%C3%A1ctil_de_usuario.

YOHAN GRATEROL. “MongoDB en Español: Tomo 1 del libro de MongoDB en español.”

Yohan Graterol, [línea], 26 de mayo de 2014.

[Consulta: 20 Agosto del 2016]

<https://www.amazon.es/MongoDB-Espa%C3%B1ol-Tomo-libro-espa%C3%B1ol-ebook/dp/B00KLGZTO0>.

ANEXOS

ANEXO A: Sprint del proyecto

Tabla 1 – A: Sprint 0 – Acciones preliminares.

Sprint 0 – Acciones Preliminares	
ID	Descripción
1	Como programador deseo obtener los requerimientos del sistema proporcionados por el usuario.
2	Como técnico deseo diseñar el modelo de la arquitectura del sistema.
3	Como programador deseo especificar un estándar de codificación.
4	Como programador deseo diseñar la base de datos para el sistema.
5	Cómo programador deseo crear las clases necesarias para correcto funcionamiento del sistema.
6	Cómo programador deseo obtener el diseño de la interfaz de usuario para el sistema

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – A: Sprint 2 – Acciones del estudiante

Sprint 2 – Acciones del estudiante	
ID	Descripción
1	Como estudiante deseo digitar mi usuario y contraseña de forma táctil.
2	Como estudiante deseo poder ingresar al sistema después de autenticarme.
3	Como estudiante deseo escoger un computador que esté disponible.
4	Como estudiante deseo que el sistema me asigne un computador de forma aleatoria de los que se encuentren disponibles.
5	Como estudiante deseo cerrar sesión en el sistema.
6	Como estudiante deseo tener un menú donde se indique la hora de inicio y tiempo de uso del computador, además de poder cerrar sesión.
7	Como estudiante deseo poder desbloquear desde el mismo computador cliente sin tener que ir a la maquina central.
8	Como administrador deseo ingresar los datos de los destinos al sistema.
9	Como administrador deseo modificar los datos de los destinos para mantenerlos actualizados en el sistema.

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – A: Sprint 3 – Administración de control y Reportes

Sprint 3 – Administración de control y Reportes	
ID	Descripción
1	El sistema debe actualizar y mostrar automáticamente las pcs que estén disponibles.
2	La app cliente y administrador deben conectarse automáticamente al servidor
3	Generar un reporte de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).
4	Generar un reporte de los estudiantes que no pueden ingresar al sistema.

Fuente: Velastegui , Lenin, 2016.

ANEXO B: Desarrollo del sprint

Anexo B.1: Pila del Sprint 0 – Acciones preliminares.

Tabla 1 – B.1: Historia de Usuario HU01-S0

HU01-S1: Como programador deseo obtener los requerimientos del sistema proporcionados por el usuario.			
Descripción	El CTE – UNACH requiere un sistema para el control de acceso de los estudiantes al laboratorio de internet, la definición de requerimientos sirve para establecer el tiempo de entrega del presente sistema y para establecer prioridades.		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	16
Criterio de Aceptación	Detallar los módulos que conformaran el sistema, así como también de las características del mismo. Planificación de proyecto.		
Tareas	T1-HU01-S0: Reunión con los directivos del CTE. T2-HU01-S0: Recolección de los requerimientos del usuario. T3-HU01-S0: Análisis de los requerimientos.		

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – B.1: Historia de Usuario HU02-S0

HU02-S0: Como técnico deseo diseñar el modelo de la arquitectura del sistema.			
Descripción	Definir la arquitectura para poder establecer las características y requerimientos del sistema.		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	10
Criterio de Aceptación	Los requerimientos desprendidos del análisis de la arquitectura del sistema deben ajustarse a los que posee la empresa.		
Tareas	T1-HU02-S0: Reunión de trabajo con los técnicos del CTE. T2-HU02-S0: Diseñar el diagrama de la arquitectura del sistema.		

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – B.1: Historia de Usuario HU03-S0

HU03-S0: Como programador deseo especificar un estándar de codificación.			
Descripción	Se debe definir un estándar para la codificación del sistema, para poder identificar fácilmente el código.		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	8
Criterio de Aceptación	Definir un estándar valido para la codificación del sistema.		
Tareas	T1-HU03-S0: Reunión de trabajo con los técnicos del CTE. T2-HU03-S0: Analizar y aplicar los estándares de codificación.		

Fuente: Velastegui , Lenin, 2016.

Tabla 4 – B.1: Historia de Usuario HU04-S0

HU04-S0: Como programador deseo diseñar la base de datos para el sistema			
Descripción	Permite realizar el diseño de la base de datos del sistema.		
Valor del negocio	10		
Puntos Estimados	24	Puntos Reales	32
Criterio de Aceptación	Diseño y análisis de la base de datos propuesta.		
Tareas	T1-HU04-S0: Reunión de trabajo con los técnicos del CTE. T2-HU04-S0: Creación de la colección usuario. T3-HU04-S0: Creación de manager para usuario.		

Fuente: Velastegui , Lenin, 2016.

Tabla 5 – B.1: Historia de Usuario HU05-S0

HU05-S0: Cómo programador deseo crear las clases necesarias para correcto funcionamiento del sistema.			
Descripción	Permite la creación de las clases para el correcto funcionamiento del sistema.		
Valor del negocio	10		
Puntos Estimados	24	Puntos Reales	32
Criterio de Aceptación	Diseñar el correcto conjunto de clases validos para el sistema		

Tareas	T1-HU05-S0: Reunión de trabajo con los técnicos del CTE T2-HU05-S0: Diseño del conjunto de clases del sistema
---------------	--

Fuente: Velastegui , Lenin, 2016.

Tabla 6 – B.1: Historia de Usuario HU06-S0

HU06-S1: Cómo programador deseo obtener el diseño de la interfaz de usuario para el sistema.			
Descripción	Permite estandarizar la interfaz de usuario para ser utilizada para cada una de las pantallas del sistema.		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	16
Criterio de Aceptación	Aprobación de colores, iconos, botones, forms etc .		
Tareas	T1-HU06-S0: Diseño de la interfaz de usuario para el sistema.		

Fuente: Velastegui , Lenin, 2016.

Anexo B.2: Pila del Sprint 1 – Acciones del Administrador.

Tabla 1 – B.2: Historia de Usuario HU02-S1

HU02-S1: Como administrador deseo digitar mi usuario y contraseña de forma táctil.			
Descripción	Permitirá que el usuario administrador del sistema pueda ingresar los datos de login de una forma táctil es decir presionando un teclado táctil presente en la pantalla.		
Valor del negocio	10		
Puntos Estimados	8	Puntos Reales	16

Criterio de Aceptación	<p>El usuario podrá ingresar su usuario y contraseña.</p> <p>Error de ingreso de datos.</p> <p>Mensaje de error campos vacíos.</p> <p>Mensaje de error validación de campos.</p>
Tareas	<p>T1-HU02-S1: Creación del método teclado táctil.</p> <p>T2-HU02-S1: Creación de la interfaz de botones táctiles.</p> <p>T3-HU02-S1: Creación de vector de botones con sus respectivos atributos.</p> <p>T4-HU02-S1: Nombre las teclas del teclado y validación de ingreso de datos.</p>

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – B.2: Historia de Usuario HU03-S1

HU03-S1: Como administrador y/o estudiante deseo tener una pantalla principal donde se muestren las opciones de ingreso al sistema.			
Descripción	Esta es la pantalla principal que se mostrara en la app administrador, esta presentara as opciones que puede realizar el usuario y también administrador.		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	16
Criterio de Aceptación	<p>Mostrar las 3 opciones:</p> <p>Administrar, elegir pc y pc aleatoria</p>		
Tareas	<p>T1-HU03-S1: Creación de la interfaz pantalla principal</p> <p>T2-HU03-S1: Programación de los botones administra, elegir pc, pc aleatorio.</p> <p>T3-HU03-S1: Al presionar alguna opción enviar al sitio correspondiente.</p> <p>T4-HU03-S1: Pruebas.</p>		

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – B.2: Historia de Usuario HU04-S1

HU04-S1: Como administrador deseo tener una interfaz donde se muestren las opciones que pueda realizar.			
Descripción	Interfaz que se mostrara al usuario administrador después de hacer login, en esta pantalla se mostraran las opciones que puede realizar el usuario administrador.		
Valor del negocio	10		
Puntos Estimados	24	Puntos Reales	48
Criterio de Aceptación	Cambio de panel Autenticación al panel administrar. Mostrar opciones lado derecho y panel de reportes en el centro. Inicializar toda la interfaz botones, textbox, combobox, etc.		
Tareas	T1-HU04-S1: Creación de los botones que realizaran la acción del administrador. T2-HU04-S1: Botón cerrar sesión y retornar al panel inicio. T3-HU04-S1: Panel para la acción de listados y reportes. T4-HU04-S1: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

Tabla 4 – B.2: Historia de Usuario HU05-S1

HU05-S1: Como administrador deseo poder desbloquear todos los computadores por un determinado tiempo ingresado de antemano.			
Descripción	Se requiere a que con dar click en el botón tiempo se desbloquearan todas las pcs durante un tiempo establecido.		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	24
Criterio de Aceptación	Mensaje de información. Mensaje de ingreso de tiempo. Mensaje de advertencia, se bloqueará por un minuto o puede cambiar de tiempo. Mensaje informativo donde se mostrará el tiempo que falta para bloquearse las pcs. Botón cambia de color verde no en uso y rojo que se activó la acción		

Tareas	<p>T1-HU05-S1: Creación de formular donde puedo ingresar el tiempo que deseo que estén desbloqueadas las pcs.</p> <p>T2-HU05-S1: creación de método temporizador.</p> <p>T3-HU05-S1: Creación de formular donde puedo se puede ver el tiempo que falta para que se bloqueen las pcs.</p> <p>T4-HU05-S1: Creación del método desbloqueo, método bloqueo pcs.</p> <p>T5-HU05-S1: Pruebas.</p>
---------------	---

Fuente: Velastegui , Lenin, 2016.

Tabla 5 – B.2: Historia de Usuario HU06-S1

HU06-S1: Como administrador deseo poder desbloquear todos los computadores hasta cuando elija bloquearlos.			
Descripción	Permitirá que desbloquear o bloquear todos los pcs disponibles con un click en la opción uso libre		
Valor del negocio	10		
Puntos Estimados	8	Puntos Reales	8
Criterio de Aceptación	Botón cambia de color verde no en uso y rojo que se activó la acción Mensaje de aceptación aceptar o cancelar		
Tareas	<p>T1-HU06-S1: Mentado cambiar de color botón.</p> <p>T2-HU06-S1: Mensaje con opciones aceptar cancelar.</p> <p>T3-HU06-S1: Cambiar estado de pcs activas a ocupadas.</p> <p>T4-HU06-S1: Desbloquear los ordenadores clientes.</p> <p>T5-HU06-S1: Pruebas.</p>		

Fuente: Velastegui , Lenin, 2016.

Tabla 6 – B.2: Historia de Usuario HU07-S1

HU07-S1: Como Administrador deseo obtener una lista de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).
--

Descripción	Permitirá mostrar una lista de los estuantes que han usado los computadores en un periodo dado por el rango de fechas, una fecha inicial y una final		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	32
Criterio de Aceptación	El usuario podrá escoger el rango de fechas. El usuario podrá escoger la opción de listado que desea mostrar. Se mostrará la lista extraída de lavase de datos en un datagridview.		
Tareas	T1-HU07-S1: Creación de la interfaz para mostrar las listas. T2-HU07-S1: Conexión con la base de datos. T3-HU07-S1: Método y select para obtener la lista. T4-HU07-S1: Llenar el datagridview con la lista optenida T5-HU07-S1: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

Tabla 7 – B.2: Historia de Usuario HU08-S1

HU08-S1: Como Administrador deseo obtener una lista de los estudiantes que no pueden ingresar al sistema.			
Descripción	Permitirá mostrar una lista de los estuantes los cuales no pueden ingresar al sistema.		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	30
Criterio de Aceptación	El usuario podrá escoger la opción de listado que desea mostrar. Se mostrará la lista extraída de lavase de datos en un datagridview		
Tareas	T1-HU07-S1: Creación de la interfaz para mostrar las listas. T2-HU07-S1: Conexión con la base de datos. T3-HU07-S1: Método y select para obtener la lista. T4-HU07-S1: Llenar el datagridview con la lista optenida T5-HU07-S1: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

Tabla 8 – B.2: Historia de Usuario HU09-S1

HU09-S1: Como administrador deseo poder bloquear o desbloquear el ingreso al sistema			
Descripción	Se podrá cambiar el campo de estado de la colección de la base de datos , permitiendo bloquear o desbloquear el ingreso al sistema		
Valor del negocio	10		
Puntos Estimados	24	Puntos Reales	32
Criterio de Aceptación	El administrador podrá cambiar el estado de un usuario. Mensaje de error campos vacíos. Mensaje de error validación de campos. Mensaje informativo ingreso exitoso.		
Tareas	T1-HU09-S1: Creación Formulario de búsqueda. T2-HU09-S1: Conexión con la base de datos. T3-HU09-S1: Consulta de usuario por medio de numero de cedula. T4-HU09-S1: Modificación de estado de usuario en bd. T5-HU09-S1: Pruebas		

Fuente: Velastegui , Lenin, 2016.

Tabla 9 – B.2: Historia de Usuario HU10-S1

HU10-S1: Como administrador deseo reiniciar todos los computadores que estén conectados a la maquina principal.			
Descripción	Esto permitirá reiniciar todas las pcs que estén conectada al servidor poder reiniciar.		
Valor del negocio	10		
Puntos Estimados	32	Puntos Reales	64
Criterio de Aceptación	El administrador podrá reiniciar todas las pcs. Mensaje de advertencia, se reiniciarán todas las pcs.		

Tareas	<p>T1-HU10-S1: Creación de mensaje con opciones.</p> <p>T2-HU10-S1: Método reiniciar pcs.</p> <p>T4-HU10-S1: Pruebas.</p>
---------------	---

Fuente: Velastegui , Lenin, 2016.

Tabla 10 – B.2: Historia de Usuario HU11-S1

HU11-S1: Como administrador deseo apagar todos los computadores que estén conectados a la maquina principal.			
Descripción	Esto permitirá reiniciar todas las pcs que estén conectada al servidor poder apagar.		
Valor del negocio	10		
Puntos Estimados	32	Puntos Reales	60
Criterio de Aceptación	El administrador podrá apagar todas las pcs. Mensaje de advertencia, se apagar todas las pcs.		
Tareas	<p>T1-HU10-S1: Creación de mensaje con opciones.</p> <p>T2-HU10-S1: Método apagar pcs.</p> <p>T4-HU10-S1: Pruebas.</p>		

Realizado por: Lenin Velastegui O.

Tabla 11 – B.2: Historia de Usuario HU12-S1

HU12-S1: Como administrador deseo cerrar sesión en el sistema y volver a la pantalla de inicio.			
Descripción	Esto permitirá salir del panel administrador y volver al panel Inicio		
Valor del negocio	10		
Puntos Estimados	16	Puntos Reales	24

Criterio de Aceptación	Se debe mostrar la pantalla de inicio.
Tareas	T1-HU10-S1: Método cerrar sesión. T2-HU10-S1: Re direccionar desde el panel administrar al inicio. T4-HU10-S1: Pruebas.

Fuente: Velastegui , Lenin, 2016.

Anexo B.3: Pila del Sprint 2 – Acciones del estudiante.

Tabla 1 – B.3: Historia de Usuario HU01-S2

HU01-S2: Como estudiante deseo digitar mi usuario y contraseña de forma táctil.			
Descripción	Permitirá que el estudiante del sistema pueda ingresar los datos de login de una forma táctil es decir presionando un teclado táctil presente en la pantalla.		
Valor del negocio	10		
Puntos Estimados	8	Puntos Reales	8
Criterio de Aceptación	El usuario podrá ingresar su usuario y contraseña. Error de ingreso de datos. Mensaje de error campos vacíos. Mensaje de error validación de campos.		
Tareas	T1-HU02-S1: Creación del método teclado táctil. T2-HU02-S1: Creación de la interfaz de botones táctiles. T3-HU02-S1: Creación de vector de botones con sus respectivos atributos. T4-HU02-S1: Nombre las teclas del teclado y validación de ingreso de datos.		

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – B.3: Historia de Usuario HU02-S2

HU02-S2: Como estudiante deseo poder ingresar al sistema después de autenticarme.			
Descripción	Permitirá mostrar la interfaz donde el estudiante podrá escoger el pc que desee de una lista de pcs que estén disponibles.		
Valor del negocio	10		
Puntos Estimados	32	Puntos Reales	24
Criterio de Aceptación	Mostrará los pc disponibles Panel con los datos del estudiante que ingreso al sistema nombre, cedula, escuela, facultad.		
Tareas	T1-HU02-S2: Creación de vector con las pcs disponibles. T2-HU02-S2: Conexión con la base de datos. T3-HU02-S2: Select que retorna los datos del estudiante. T4-HU02-S2: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – B.3: Historia de Usuario HU03-S2

HU03-S2: Como estudiante deseo escoger un computador que esté disponible.			
Descripción	El estudiante podrá escoger escoger la pc que desee.		
Valor del negocio	10		
Puntos Estimados	48	Puntos Reales	72
Criterio de Aceptación	El sistema mostrara con color azul claro las pcs disponibles. El sistema mostrara con color azul oscuro las pcs no disponibles. Mensaje ya se encuentra en uso.		
Tareas	T1-HU03-S2: Creación del método escoger pc. T2-HU03-S2: conexión con base de datos. T3-HU03-S2: Cerrar sesión T4-HU03-S2: Método escoger Pc. T5-HU03-S2: Desbloquear pc escogida. T4-HU03-S2: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

Tabla 4 – B.3: Historia de Usuario HU04-S2

HU04-S2: Como estudiante deseo que el sistema me asigne un computador de forma aleatoria de los que se encuentren disponibles			
Descripción	Metodo por el cual el sistema podrá asignar una pc de forma aleatoria.		
Valor del negocio	10		
Puntos Estimados	15	Puntos Reales	15
Criterio de Aceptación	Mensaje de error campos vacíos. Mensaje de error validación de campos. Mensaje informativo pc no disponibles.		
Tareas	T1-HU04-S2: Creación del método asignación aleatoria. T2-HU04-S2: Creación de la interfaz pc aleatoria. T3-HU04-S2: Validaciones. T4-HU04-S2: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

Tabla 5 – B.3: Historia de Usuario HU05-S2

HU05-S2: Como estudiante deseo cerrar sesión en el sistema.			
Descripción	Interfaz por la que el estudiante podrá cerrar sesión y salir de la pantalla actual a la pantalla de inicio		
Valor del negocio	10		
Puntos Estimados	32	Puntos Reales	64
Criterio de Aceptación	Mensaje de error campos vacíos. Mensaje de error validación de campos.		
Tareas	T1-HU05-S2: Creación método salir. T2-HU05-S2: Salir pantalla elegir pc a pantalla inicio. T3-HU05-S2: Validaciones. T4-HU05-S2: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

Tabla 6 – B.3: Historia de Usuario HU06-S2

HU06-S2: Como estudiante deseo tener un menú donde se indique la hora de inicio y tiempo de uso del computador, además de poder cerrar sesión.			
Descripción	Panel pequeño que se mostrara después de desbloquear la pc en el lado del cliente, el panel mostrara el nombre del usuario hora de inicio y tiempo de uso del pc.		
Valor del negocio	10		
Puntos Estimados	24	Puntos Reales	48
Criterio de Aceptación	Mensaje de error campos vacíos. Mensaje de error validación de campos.		
Tareas	T1-HU06-S2: Creación control sesión. T2-HU06-S2: Creación método cronometro. T3-HU06-S2: Conexión base de datos. T4-HU06-S2: Select para mostrar nombre de usuario. T5-HU06-S2: Guardar fecha y hora de cierre de sesión. T6-HU06-S2: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

Tabla 7 – B.3: Historia de Usuario HU07-S2

HU07-S2: Como estudiante deseo poder desbloquear desde el mismo computador cliente sin tener que ir a la maquina central.			
Descripción	Podra hacer login desde la app cliente y guarda los datos uso del computador		
Valor del negocio	10		
Puntos Estimados	48	Puntos Reales	40
Criterio de Aceptación	Mensaje de error campos vacíos. Mensaje de error validación de campos.		

Tareas	<p>T1-HU07-S2: Creación del método login.</p> <p>T2-HU07-S2: Conexión con base datos.</p> <p>T3-HU07-S2: Consulta a ver si existen los datos ingresados.</p> <p>T4-HU07-S2: Pruebas.</p>
---------------	--

Fuente: Velastegui , Lenin, 2016.

Anexo B.4: Pila del Sprint 3 – Administración de control y Reportes.

Tabla 1 – B.4: Historia de Usuario HU01-S3

HU01-S3: El sistema debe actualizar y mostrar automáticamente las pcs que estén disponibles.			
Descripción	El sistema visualizara las pcs que estén disponibles (pc conectadas) y las borrara cuando las pcs se desconecten		
Valor del negocio	10		
Puntos Estimados	32	Puntos Reales	70
Criterio de Aceptación	<p>Mensaje de error campos vacíos.</p> <p>Mensaje de error validación de campos.</p>		
Tareas	<p>T1-HU01-S3: Creación del método Conexión.</p> <p>T2-HU01-S3: Creación método desconexion.</p> <p>T3-HU01-S3: Validaciones.</p> <p>T4-HU01-S3: Pruebas.</p>		

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – B.4: Historia de Usuario HU02-S3

HU02-S3: La app cliente y administrador deben conectarse automáticamente al servidor.			
Descripción	Se requiere que la app cliente debe estar corriendo en segundo plano esperando a que se encienda la app administrador para que se bloquee automáticamente la pc		
Valor del negocio	10		
Puntos Estimados	40	Puntos Reales	32

Criterio de Aceptación	<p>Mensaje de error campos vacíos.</p> <p>Mensaje de error validación de campos.</p> <p>Mensaje informativo conexión exitosa.</p> <p>Mensaje informativo error conexión.</p>
Tareas	<p>T1-HU02-S3: Creación del método conexión server.</p> <p>T2-HU02-S3: Creación envió de mensajes por socket.</p> <p>T3-HU02-S3: Validaciones.</p> <p>T4-HU02-S3: Pruebas.</p>

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – B.4: Historia de Usuario HU03-S3

HU03-S3: Generar un reporte de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).			
Descripción	Sistema permite generar reporte de los estudiantes que han usado las pcs en un determinado tiempo fechas.		
Valor del negocio	10		
Puntos Estimados	42	Puntos Reales	42
Criterio de Aceptación	<p>El usuario podrá escoger el rango de fechas.</p> <p>Se mostrará la lista extraída de la base de datos en un crystalReportViewer1</p>		
Tareas	<p>T1-HU07-S1: Creación de la interfaz para mostrar las listas.</p> <p>T2-HU07-S1: Conexión con la base de datos.</p> <p>T3-HU07-S1: Método y select para obtener la lista.</p> <p>T4-HU07-S1: Llenar el crystalReportViewer1 con la lista obtenida</p> <p>T5-HU07-S1: Pruebas.</p>		

Fuente: Velastegui , Lenin, 2016.

Tabla 4 – B.4: Historia de Usuario HU04-S3

HU04-S3: Generar un reporte de los estudiantes que no pueden ingresar al sistema..			
Descripción	Permitirá mostrar una lista de los estuantes los cuales no pueden ingresar al sistema.		
Valor del negocio	10		
Puntos Estimados	32	Puntos Reales	32
Criterio de Aceptación	El usuario podrá escoger la opción de listado que desea mostrar. Se mostrará la lista extraída de lavase de datos en un datagridview		
Tareas	T1-HU07-S1: Creación de la interfaz para mostrar las listas. T2-HU07-S1: Conexión con la base de datos. T3-HU07-S1: Método y select para obtener la lista. T4-HU07-S1: Llenar el datagridview con la lista optenida T5-HU07-S1: Pruebas.		

Fuente: Velastegui , Lenin, 2016.

ANEXO C: Diccionario de datos

Tabla 1 – C: Diccionario de datos de las colecciones

Collection	Campo	Tipo de Dato	PK	FK
Tipo Usuario	Id_tipousuario	ObjectID	x	
	Descripcion	Texto		
Facultad	Id_facultad	ObjectID	x	
	Descripcion	Texto		
Escuela	Id_escuela	ObjectID	x	
	Descripcion	Texto		
Usuarios	Id_facultad	ObjectID		x
	Id_usuarios	ObjectID	x	
	Ci	Texto		
	Nombre	Texto		
	Estado	Texto		

	Sexo	Texto		
	Contraseña	Texto		
	Nick	Texto		
	Id_escuela	ObjectID		x
	Id_tipousuario	ObjectID		x
RegistroUso	Id_registrouso	ObjectID	x	
	Id_usuarios	ObjectID		x
	Hora_fecha_uso	DateTime		
	Tiempo_uso	Texto		
	numPc	Int		
	Hora_fecha_uso	DateTime		

Fuente: Velastegui , Lenin, 2016.

ANEXO D: Pruebas de Aceptación

Anexo D.1: Pruebas de aceptación Pila del Sprint 0 – Actividades iniciales

Tabla 1 – D.1: Prueba requerimientos del sistema

Ficha de prueba	
Prueba – 01: Requerimientos del sistema	
Código de HU: HU01-S0	
Descripción de HU: Como programador deseo obtener los requerimientos del sistema para definir sus funcionalidades.	
Estado: Superado	
Evento	Valor Esperado
Reunión con personas involucradas en el desarrollo del sistema.	Definición de características del sistema con todos sus módulos a desarrollar.
Presentación de propuesta del sistema	Definición de tecnología y recursos a utilizar.
Planificación del sistema	Definición de la planificación.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – D.1: Prueba arquitectura del sistema

Ficha de prueba	
Prueba – 02: Arquitectura del sistema	
Código de HU: HU02-S0	
Descripción de HU: Como técnico deseo obtener un modelo para la arquitectura del sistema para establecer las necesidades de hardware y software.	
Estado: Superado	
Evento	Valor Esperado
Propuesta de sistema. Plantación de arquitectura en tres capas	Arquitectura del sistema correspondiente al sistema
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – D.1: Prueba estándar de codificación del sistema

Ficha de prueba	
Prueba – 03: Estándar de codificación del sistema	
Código de HU: HU03-S0	
Descripción de HU: Como programador deseo obtener un estándar de codificación del proyecto para mantener una escritura fija en el proyecto	
Estado: Superado	
Evento	Valor Esperado
Revisión de estándares de codificación del sistema	Definición de los parámetros para ser utilizados al desarrollar el sistema.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 4 – D.1: Prueba diseño de base de datos.

Ficha de prueba	
Prueba – 04: Diseño de base de datos.	
Código de HU: HU04-S0	
Descripción de HU: Como programador deseo obtener el diseño de la base de datos.	
Estado: Superado	
Evento	Valor Esperado
Revisión del modelo relacional de la base de datos	Modelo relacional de la base de datos.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 5 – D.1: Prueba distribución de clases del sistema

Ficha de prueba	
Prueba – 05: Distribución de clases del sistema.	
Código de HU: HU05-S0	
Descripción de HU: Cómo programador deseo obtener la distribución correcta de clases del sistema.	
Estado: Superado	
Evento	Valor Esperado
Revisión del diagrama de clases del sistema.	Diagrama de clases del sistema.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 6 – D.1: Prueba diseño de la interfaz de usuario del sistema

Ficha de prueba	
Prueba – 06: Diseño de la interfaz de usuario del sistema.	
Código de HU: HU06-S0	

Descripción de HU: Cómo programador deseo obtener el diseño de la interfaz de usuario para el sistema.	
Estado: Superado	
Evento	Valor Esperado
Presentación de bosquejo de pantallas a los usuarios.	Aprobación de los prototipos presentados.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Anexo D.2: Pruebas de aceptación Pila del Sprint 1 – Administración de pedidos

Tabla 1 – D.2: Ingresar al sistema a través de un username y contraseña.

Ficha de prueba	
Prueba – 01: Ingresar al sistema a través de un username y contraseña.	
Código de HU: HU02-S1	
Descripción de HU: Como administrador deseo poder ingresar al sistema a través de un username y contraseña.	
Estado: Superado	
Evento	Valor Esperado
Ingresar campos vacíos. Error de conexión a la BD. Botón aceptar.	Mensaje de error: campos vacíos. Mensaje de error: error de conexión. Mensaje de confirmación: bienvenida nombre usuario.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – D.2: Digitar mi usuario y contraseña de forma táctil.

Ficha de prueba	
Prueba – 01: Digitar mi usuario y contraseña de forma táctil	
Código de HU: HU03-S1	

Descripción de HU: Como administrador deseo digitar mi usuario y contraseña de forma táctil.	
Estado: Superado	
Evento	Valor Esperado
Ingresar campos vacíos. Error de conexión a la BD. Mostrar teclado táctil.	Mensaje de error: campos vacíos. Mensaje de error: error de conexión. Visualización y tipeo de teclado táctil.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – D.2: Tener una pantalla principal donde se muestren las opciones de ingreso al sistema.

Ficha de prueba	
Prueba – 01: Tener una pantalla principal donde se muestren las opciones de ingreso al sistema.	
Código de HU: HU04-S1	
Descripción de HU: Como administrador y/o estudiante deseo tener una pantalla principal donde se muestren las opciones de ingreso al sistema.	
Estado: Superado	
Evento	Valor Esperado
Ejecución del sistema. Pantalla principal.	Visualización de la pantalla principal. Mostrar botones con opciones cliente/administrador.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 4 – D.2: Tener una interfaz donde se muestren las opciones que pueda realizar.

Ficha de prueba	
Prueba – 01: Tener una interfaz donde se muestren las opciones que pueda realizar.	
Código de HU: HU05-S1	

Descripción de HU: Como administrador deseo tener una interfaz donde se muestren las opciones que pueda realizar.	
Estado: Superado	
Evento	Valor Esperado
Botón aceptar. Panel y opciones administrador. Creación de forms.	Inicializar componentes. Actualización de form. Mostrar forms.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 5 – D.2: Desbloquear todos los computadores por un determinado tiempo ingresado de antemano.

Ficha de prueba	
Prueba – 01: Desbloquear todos los computadores por un determinado tiempo ingresado de antemano.	
Código de HU: HU06-S1	
Descripción de HU: Como administrador deseo poder desbloquear todos los computadores por un determinado tiempo ingresado de antemano.	
Estado: Superado	
Evento	Valor Esperado
Desbloquear por tiempo. Aceptar tiempo. Temporizador Terminar tiempo	Pedir tiempo. Mostrar mensaje de aviso. Mostrar tiempo que falta para bloquear pcs Bloquean todas las pcs
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 6 – D.2: Desbloquear todos los computadores hasta cuando elija bloquearlos.

Ficha de prueba	
Prueba – 01: Desbloquear todos los computadores hasta cuando elija bloquearlos.	
Código de HU: HU07-S1	
Descripción de HU: Como administrador deseo poder desbloquear todos los computadores hasta cuando elija bloquearlos.	
Estado: Superado	
Evento	Valor Esperado
Desbloquear pcs. Bloquear Pcs Descbloquear Pcs.	Quita el bloqueo de las pcs cliente. Pone pantalla de bloqueo en pcs clientes. Cambian de estado las pcs de disponibles a ocupados
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 7 – D.2: Obtener una lista de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).

Ficha de prueba	
Prueba – 01: Obtener una lista de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).	
Código de HU: HU08-S1	
Descripción de HU: Como Administrador deseo obtener una lista de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).	
Estado: Superado	
Evento	Valor Esperado
Filtrar la lista de usuarios por un rango de fechas una inicial y una final Conexión al BD	Muestra la lista de estudiantes que usaron las pcs en ese determinado tiempo. Mensaje de confirmación
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 8 – D.2: Obtener una lista de los estudiantes que no pueden ingresar al sistema.

Ficha de prueba	
Prueba – 01: Obtener una lista de los estudiantes que no pueden ingresar al sistema.	
Código de HU: HU09-S1	
Descripción de HU: Como Administrador deseo obtener una lista de los estudiantes que no pueden ingresar al sistema.	
Estado: Superado	
Evento	Valor Esperado
Filtrar la lista de usuarios por estado baneados	Muestra la lista con estudiantes que no puedan ingresar al sistema
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 9 – D.2: Bloquear o desbloquear el ingreso al sistema.

Ficha de prueba	
Prueba – 01: Bloquear o desbloquear el ingreso al sistema	
Código de HU: HU10-S1	
Descripción de HU: Como administrador deseo poder bloquear o desbloquear el ingreso al sistema.	
Estado: Superado	
Evento	Valor Esperado
Ingresar campos vacíos.	Mensaje de error: campos vacíos.
Error de conexión a la BD.	Mensaje de error: error de conexión.
Modificación del estado usuario.	Mensaje de confirmación: modificación exitosa.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 10 – D.2: Reiniciar todos los computadores que estén conectados a la maquina principal.

Ficha de prueba	
Prueba – 01: Reiniciar todos los computadores que estén conectados a la maquina principal.	
Código de HU: HU10-S1	
Descripción de HU: Como administrador deseo reiniciar todos los computadores que estén conectados a la maquina principal.	
Estado: Superado	
Evento	Valor Esperado
Reiniciar Tiempo	Mensaje de advertencia: pc se apagara Tiempo que se va a demorar antes de reiniciar
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 9 – D.2: Apagar todos los computadores que estén conectados a la maquina principal.

Ficha de prueba	
Prueba – 01: Reiniciar todos los computadores que estén conectados a la maquina principal.	
Código de HU: HU10-S1	
Descripción de HU: Como administrador deseo apagar todos los computadores que estén conectados a la maquina principal.	
Estado: Superado	
Evento	Valor Esperado
Apagar Tiempo	Mensaje de advertencia: pc se apagara Tiempo que se va a demorar antes de apagara
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Anexo D.3: Pruebas de aceptación Pila del Sprint 1 – Acciones del estudiante

Tabla 1 – D.3: Digitar mi usuario y contraseña de forma táctil.

Ficha de prueba	
Prueba – 01: Digitar mi usuario y contraseña de forma táctil.	
Código de HU: HU01-S1	
Descripción de HU: Como estudiante deseo digitar mi usuario y contraseña de forma táctil.	
Estado: Superado	
Evento	Valor Esperado
Datos Vacíos. Botón aceptar.	Mensaje de error: Datos no válidos. Ingreso al menú de Usuario
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – D.3: Prueba ingresar al sistema después de autenticarme.

Ficha de prueba	
Prueba – 01: Ingresar al sistema después de autenticarme.	
Código de HU: HU02-S1	
Descripción de HU: Como estudiante deseo poder ingresar al sistema después de autenticarme.	
Estado: Superado	
Evento	Valor Esperado
Inicializar componentes. Cargar botones con las pcs disponibles Diseñar página del usuario.	Mostrar componentes Muestra botones con las acciones. Mostrar página del usuario.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – D.3: Escoger un computador que esté disponible.

Ficha de prueba	
Prueba – 01: Escoger un computador que esté disponible.	
Código de HU: HU03-S1	
Descripción de HU: Como estudiante deseo escoger un computador que esté disponible.	
Estado: Superado	
Evento	Valor Esperado
Escoger Pc Error de conexión de la BD. Pc Escogida. Pc Seleccionada.	Mensaje de Advertencia: Desea escoger este pc. Mensaje de error: error de conexión. Cambiar de estado al botón pc escogido. Mensaje Información: La Pc ya está en uso.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 4 – D.3: Asignar un computador de forma aleatoria de los que se encuentren disponibles.

Ficha de prueba	
Prueba – 01: Asignar un computador de forma aleatoria de los que se encuentren disponibles.	
Código de HU: HU04-S1	
Descripción de HU: Como estudiante deseo que el sistema me asigne un computador de forma aleatoria de los que se encuentren disponibles.	
Estado: Superado	
Evento	Valor Esperado
No asignar pc. Asignar Pc.	Mensaje de error: No hay pc disponibles. Mensaje de éxito: Pc # fue asignada.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 5 – D.3: Cerrar sesión en el sistema.

Ficha de prueba	
Prueba – 01: Cerrar sesión en el sistema.	
Código de HU: HU05-S1	
Descripción de HU: Como estudiante deseo cerrar sesión en el sistema.	
Estado: Superado	
Evento	Valor Esperado
Cerrar Sesión. Cerrar sesión.	Guardar datos en la BD. Cerrar form control sesión y maximizar form client.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 6 – D.3: Tener un menú donde se indique la hora de inicio y tiempo de uso del computador, además de poder cerrar sesión.

Ficha de prueba	
Prueba – 01: Tener un menú donde se indique la hora de inicio y tiempo de uso del computador, además de poder cerrar sesión.	
Código de HU: HU06-S1	
Descripción de HU: Tener un menú donde se indique la hora de inicio y tiempo de uso del computador, además de poder cerrar sesión.	
Estado: Superado	
Evento	Valor Esperado
Escoger Pc Escoger PC.	Mostrar Form control sesión con los datos del usuario. Guardar datos en la base de datos
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 7 – D.3: Desbloquear desde el mismo computador cliente sin tener que ir a la maquina central.

Ficha de prueba	
Prueba – 01: Desbloquear desde el mismo computador cliente sin tener que ir a la maquina central.	
Código de HU: HU07-S1	
Descripción de HU: Como estudiante deseo poder desbloquear desde el mismo computador cliente sin tener que ir a la maquina central.	
Estado: Superado	
Evento	Valor Esperado
Campos vacíos. Presionar botón Ingresar. Presionar botón Ingresar.	Mensaje de error: Datos no válidos. Mensaje éxito: bienvenido nombre usuario. Guardar datos en la bd.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Anexo D.4: Pruebas de aceptación Pila del Sprint 3 – Administración de control y Reportes.

Tabla 1 – D.4: Actualizar y mostrar automáticamente las pcs que estén disponles

Ficha de prueba	
Prueba – 01: Actualizar y mostrar automáticamente las pcs que estén disponles.	
Código de HU: HU01-S3	
Descripción de HU: El sistema debe actualizar y mostrar automáticamente las pcs que estén disponles.	
Estado: Superado	
Evento	Valor Esperado
Conenctar Pcs. Desconectar Pcs	Mostrar en el panel como pcs disponibles. Desaparecer en el panel como pcs.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 2 – D.4: Conectarse automáticamente al servidor

Ficha de prueba	
Prueba – 01: Conectarse automáticamente al servidor	
Código de HU: HU02-S3	
Descripción de HU: La app cliente y administrador deben conectarse automáticamente al servidor	
Estado: Superado	
Evento	Valor Esperado
Ejecutar app cliente Espera mensaje prendido servidor	Espera hasta que se encienda el servidor Se bloquea el cliente.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 3 – D.4: Generar un reporte de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).

Ficha de prueba	
Prueba – 01: Generar un reporte de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).	
Código de HU: HU03-S3	
Descripción de HU: Generar un reporte de los usuarios que usaron los computadores en un determinado periodo de tiempo (fechas).	
Estado: Superado	
Evento	Valor Esperado
Búsqueda de la lista	Mostrar la lista y posteriormente generar el reporte de los estudiantes que usaron las pcs durante ese tiempo.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.

Tabla 4 – D.4: Generar un reporte de los estudiantes que no pueden ingresar al sistema.

Ficha de prueba	
Prueba – 01: Generar un reporte de los estudiantes que no pueden ingresar al sistema.	
Código de HU: HU04-S3	
Descripción de HU: Generar un reporte de los estudiantes que no pueden ingresar al sistema.	
Estado: Superado	
Evento	Valor Esperado
Listado de todos los usuarios bloqueados del sistema	Mostrar un listado de todos los usuarios que no pueden ingresar al sistema para posteriormente generar el reporte.
Observaciones:	

Fuente: Velastegui , Lenin, 2016.