

**SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE ESPACIOS, RECURSOS
Y SERVICIOS DE LA UNIDAD DE INFORMÁTICA Y TELECOMUNICACIONES
(UIT) DE LA UNIVERSIDAD DE NARIÑO**

ALBEIRO NICOLÁS CHAMORRO PORTILLA

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2014**

**SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE ESPACIOS, RECURSOS
Y SERVICIOS DE LA UNIDAD DE INFORMÁTICA Y TELECOMUNICACIONES
(UIT) DE LA UNIVERSIDAD DE NARIÑO**

ALBEIRO NICOLÁS CHAMORRO PORTILLA

**Trabajo de Grado presentado como requisito parcial para optar al título de
Ingeniero de Sistemas**

**Director
JESÚS H. INSUASTI PORTILLA, MSc**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2014**

NOTA DE RESPONSABILIDAD

“Las ideas y conclusiones aportadas en la tesis de grado, son responsabilidad exclusiva de sus autores”.

Artículo 1 del acuerdo No 324 del 11 de octubre de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”

Artículo 13, Acuerdo N. 005 de 2010 emanado del Honorable Consejo Académico

NOTA DE ACEPTACIÓN

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Firma del Director de proyecto

San Juan de Pasto, noviembre de 2014

AGRADECIMIENTOS

Al Ing. Jesús H. Insuasti, MSc, por su gran colaboración, apoyo, dedicación, consejos y sobre todo, por su confianza en mí.

A todos los funcionarios de la UIT, por la dedicación y el empeño en la realización de este proyecto, por su información precisa y oportuna.

A mis compañeros de trabajo en la OASI, por enseñarme que trabajar en lo que a uno le gusta, siempre es placentero, también por alegrarme la vida en los momentos difíciles.

A la Universidad de Nariño por brindarme tantos espacios y oportunidades de acrecentar mis conocimientos.

A los docentes de la Universidad de Nariño, quienes dentro y fuera de clase, pusieron sus conocimientos a disposición, haciendo posible que mi formación profesional sea la más completa posible, para estar preparado en el mundo laboral.

A mis compañeros de estudio con los cuales compartí y construí grandes momentos.

DEDICATORIA

Este trabajo está dedicado a todas esas personas que estuvieron conmigo desde el inicio y también a las que fueron llegando a mi vida con el pasar del tiempo, a quienes me apoyaron para iniciar este camino y que me dieron fuerzas para continuar cuando las circunstancias eran difíciles, a todas esas personas que de una u otra manera creyeron en mí, que confiaron en mis capacidades y mis deseos de salir adelante, para ellos toda la gratitud del mundo.

Una mención y una dedicatoria especial para mis padres, mis hermanas y mi novia, ya que me brindaron la fuerza y la voluntad para llevar esta gran meta a feliz término, espero poder corresponder a sus grandes esfuerzos.

ALBEIRO N. CHAMORRO P.

MARCAS REGISTRADAS

Microsoft SQL Server 2012, Microsoft Visual Studio 2012 y Microsoft Windows, son marcas o marcas registradas de Microsoft Corporation.

Programación Extrema© Copyright (c) 1999, 2000, 2001, 2004, 2009 Es producto creado por Donovan Wells.

PHP (Hypertext Pre-processor o inicialmente PHP Tools, o, Personal Home Page Tools) Fue creado por Rasmus Lerdorfy pertenece a The PHP Group.

Otros nombres propios pueden ser marcas o marcas registradas y son propiedad de sus respectivos propietarios.

RESUMEN

Este proyecto surgió debido a la necesidad que presentaba la UIT de tener un sistema integrado para el manejo de todos sus procesos, facilitando la organización de la información y el seguimiento de los equipos sobre los cuales se tiene responsabilidad.

En este documento, primero se presenta una descripción detallada de las generalidades del proyecto, fragmentando la idea de lo que se desea para su fácil comprensión. Luego se obtuvo una breve y simplificada información de la organización y algunos procesos manejados en la UIT. Posterior a esto, se hizo una descripción de las herramientas que se utilizaron para la implementación del sistema de información integrado. También se tiene la explicación de la metodología que se usó, para que el desarrollo del proyecto se hiciera de una manera organizada, garantizando un producto final de alta calidad.

El siguiente paso fue colocar en práctica los pasos descritos en la metodología, para poner en marcha el proyecto, obteniendo requerimientos, diseñando módulos, desarrollando componentes, testeando los mismos y desplegando el sistema para el uso de los usuarios finales.

El proyecto concluyó con la debida capacitación a los funcionarios de la UIT, para su adecuada utilización.

ABSTRACT

This project was born due the necessity of at UIT to have an integrated management system for its processes, helping the organization of information, and the monitoring of the equipment which is under responsibility of this area.

In this paper, first is presented a detailed description of this project splitting the purpose for an easy understanding. A brief and simplified information of the organization and some processes managed in the UIT is later describe. Following this is showed a description of the tools used for the implementation of integrated information system developed. There is also an explanation of the methodology used, so that the project was done in an organized manner, ensuring a high quality end product.

The next phase was apply the steps define in the methodology to implement the project , obtaining requirements, designing modules, components developing, test and deploying the same system for use by end users.

The project concluded with training to administrative personal at the UIT for correct use of system

CONTENIDO

	Pág.
GLOSARIO	14
INTRODUCCIÓN	15
1. MARCO TEÓRICO	16
1.1 UIT	16
1.1.1 Áreas.	16
1.1.2 Servicios.	16
1.2 SISTEMA GESTOR DE BASES DE DATOS MS SQLSERVER 2012	17
1.2.1 Características.	17
1.2.2 T-SQL.	18
1.3 IDE MICROSOFT VISUAL STUDIO 2012	19
1.3.1 Novedades de Ms Visual Studio 2013.	19
1.4 NET FRAMEWORK	28
1.4.1 Objetivos del .Net Framework.	29
1.4.2 Novedades de .Net Framework 4.5.1.	30
1.5 PHP	30
1.5.1 Características de PHP.	30
2. METODOLOGÍA	32
2.1 METODOLOGÍAS DE DESARROLLO DE SOFTWARE	32
2.2 EL NUEVO ENFOQUE	33
2.3 METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE	34
2.3.1 Extreme Programming (Programación Extrema).	35
3. SISTEMA DE INFORMACIÓN PARA EL MANEJO DE RECURSOS DE LA UNIDAD DE INFORMÁTICA Y TELECOMUNICACIONES DE LA UNIVERSIDAD DE NARIÑO	38
3.1 FASE DE PLANEACIÓN	38
3.1.1 Historias de usuario.	38
3.2 PLAN DE ENTREGAS	43
3.3 FASE DE DISEÑO	44
3.3.1 Módulo de usuarios.	44
3.3.2 Módulo de solicitudes.	45
3.3.3 Módulo de inventario.	46
3.4.4 Módulo de horarios de monitores.	46
3.4.5 Módulo de aulas de informática.	47
3.4.6 Módulo de reservas y préstamos de equipos.	47
3.4.7 Módulo de reportes.	47
3.5 DISEÑO DE LA BASE DE DATOS	48
3.5.1 Detalles tabla aulas.	49

3.5.2	Detalles tabla dependencias	50
3.5.3	Detalles tabla dispositivos equipo.	51
3.5.4	Detalles tabla equipos.	52
3.5.6	Detalles tabla horarios aulas.	55
3.5.7	Detalles tabla horariosm.	56
3.5.8	Detalles tabla mantenimientos equipos.	57
3.5.9	Detalles tabla permisos.	58
3.5.10	Detalles tabla préstamos.	59
3.5.11	Detalles tabla préstamosotros.	60
3.5.12	Detalles tabla programas.	61
3.5.12	Detalles tabla recursos.	62
3.5.13	Detalles tabla roles.	62
3.5.14	Detalles tabla rolespermisorecursos	63
3.5.15	Detalles tabla sedes.	64
3.5.16	Detalles tabla softwareaulas.	64
3.5.17	Detalles tabla solicitudes.	65
3.5.18	Detalles tabla ubicación.	67
3.5.19	Detalles tabla usuarios.	68
3.6	TARJETAS CRC	69
3.6.1	Tarjeta CRC menú principal.	69
3.6.2	Tarjeta CRC módulo usuarios.	69
3.6.3	Tarjeta CRC módulo solicitudes	70
3.6.4	Tarjeta CRC módulo inventario.	70
3.6.5	Tarjeta CRC módulo horarios monitores.	70
3.6.6	Tarjeta CRC módulo aulas	71
3.6.7	Tarjeta CRC módulo préstamos.	71
3.6.8	Tarjeta CRC módulo reportes.	72
3.7	FASE DE PRUEBAS	72
3.7.1	Prueba de aceptación: Hu1- acceso al sistema.	73
3.7.2	Prueba de aceptación: Hu2 - usuarios y roles.	74
3.7.3	Prueba de aceptación: HU3 - crear solicitudes.	75
3.7.4	Prueba de aceptación: HU4 - reserva de equipos.	76
3.7.5	Prueba de aceptación: Hu5 - inventario de equipos.	78
3.7.6	Prueba de aceptación: HU6 – mantenimientos.	80
3.7.7	Prueba de aceptación: HU7 – solicitudes.	81
3.7.8	Prueba de aceptación: HU8 - horarios monitores.	83
3.7.9	Prueba de aceptación: HU9 – aulas.	85
3.7.10	Prueba de aceptación: HU10 – prestamos.	87
3.7.11	Prueba de aceptación: HU11 – reportes.	89
4.	CONCLUSIONES	91
5.	RECOMENDACIONES	92
	BIBLIOGRAFÍA	93

LISTA DE TABLAS

	pág.
Tabla 1. Acceso al Sistema	38
Tabla 2. Usuarios y Roles	39
Tabla 3. Crear, Calificar y Revisar el Historial de Solicitudes	39
Tabla 4. Reserva de Equipos	40
Tabla 5. Inventario de Equipos	40
Tabla 6. Mantenimientos	41
Tabla 7. Historia de Usuario Solicitudes	41
Tabla 8. Horarios Monitores	42
Tabla 9. Aulas	42
Tabla 10. Préstamos	43
Tabla 11. Reportes	43
Tabla 12. Plan de Entregas	44
Tabla 13. Tablas de la Base de Datos	48
Tabla 14. Tarjeta CRC menú principal	69
Tabla 15. Tarjeta CRC módulo usuarios	69
Tabla 16. Tarjeta CRC módulo solicitudes	70
Tabla 17. Tarjeta CRC módulo inventario	70
Tabla 18. Tarjeta CRC módulo horarios monitores	70
Tabla 19. Tarjeta CRC módulo aulas	71
Tabla 20. Tarjeta CRC módulo préstamos	71
Tabla 21. Tarjeta CRC módulo reportes	72
Tabla 22. Prueba de aceptación: Hu1- acceso al sistema	73
Tabla 23. Prueba de aceptación: Hu2 - usuarios y roles	74
Tabla 24. Prueba De aceptación: HU3 - crear solicitudes	75
Tabla 25. Prueba de aceptación: HU4 - reserva de equipos	76
Tabla 26. Prueba de aceptación: HU5 - inventario de equipos	78
Tabla 27. Prueba de aceptación: HU6 – mantenimientos	80
Tabla 28. Prueba de aceptación: HU7 – solicitudes	81
Tabla 29. Prueba de aceptación: HU8 - horarios monitores	83
Tabla 30. Prueba de aceptación: HU9 – aulas	85
Tabla 31. Prueba de aceptación: HU10 – prestamos	87
Tabla 32. Prueba de aceptación: HU11 - REPORTES	89

LISTA DE FIGURAS

	Pág.
Figura 1. Áreas UIT	16
Figura 2. Windows Store	20
Figura 3. Interfaz de Blend	21
Figura 4. Tipos de Proyectos	22
Figura 5. Ejemplo de Analizador de Rendimiento	23
Figura 6. Agile Portfolio Management	25
Figura 7. Team Rooms	26
Figura 8. Team Foundation Server	27
Figura 9. Gráficos con Team Foundation Server	28
Figura 10. Estructura de .NET Framework	29
Figura 11. Grafica Costo-Tiempo Modelo en Cascada o Modelo Incremental	34
Figura 12. Gráfica Costo-Tiempo Metodologías Agiles de Desarrollo	35
Figura 13. Esquema de Desarrollo en XP	37
Figura 14. Detalles tabla aulas	49
Figura 15. Detalles tabla dependencias	50
Figura 16. Detalles tabla dispositivos equipo	51
Figura 17. Detalles tabla equipos	52
Figura 18. Detalles tabla horarios aulas	55
Figura 19. Detalles tabla horariosm	56
Figura 20. Detalles tabla mantenimientos equipos	57
Figura 21. Detalles tabla permisos	58
Figura 22. Detalles tabla préstamos	59
Figura 23 Detalles tabla préstamosotros	60
Figura 24. Detalles tabla programas	61
Figura 25. Detalles tabla recursos	62
Figura 26. Detalles tabla roles	62
Figura 27. Detalles tabla rolespermisorecursos	63
Figura 28. Detalles tabla sedes	64
Figura 29. Detalles tabla softwareaulas	64
Figura 30. Detalles tabla solicitudes	65
Figura 31. Detalles tabla ubicación	67
Figura 32. Detalles tabla usuarios	68

GLOSARIO

CLR: El primer componente de .NET que conoceremos es el Common Language Runtime, también denominado CLR. Este es un programa de ejecución común a todos los lenguajes. Este programa se encarga de leer el código generado por el compilador y empieza su ejecución. Sin importar si el programa fue creado con C#, con Visual Basic .NET o algún otro lenguaje de .NET el CLR lo lee y ejecuta.

EQUIPO: Todos los componentes de cómputo y afines sobre los cuales la UIT tiene algún tipo de responsabilidad.

FRAMEWORK: Es la integración de herramientas para el desarrollo de software, con la finalidad de incrementar sus funcionalidades.

IDE (Integrated Developer Environment): Facilita la producción de sistemas de información, ya que trae las herramientas necesarias para trabajar sin tener que usar varias aplicaciones para el manejo de los diferentes recursos.

MONITOR: estudiante de la universidad, que laboran en la UIT, ya sea en modalidad técnico o medio tiempo y que tiene interacción directa con los equipos y muchos de los procesos que se manejan en la dependencia.

SISTEMA: Conjunto de partes o componentes que interactúan entre sí, para alcanzar un objetivo.

SISTEMA DE INFORMACIÓN: Un sistema de información, son herramientas que se organizan para tratar datos y generar información de calidad y de interés para el usuario final del sistema.

USUARIO FINAL: son todas las personas que tendrán interacción con el sistema de información, introduciendo datos u obteniendo resultados generados por el mismo.

UIT: Unidad de Informática y Telecomunicaciones o más conocida como Aula de Informática, es la encargada de administrar los recursos informáticos y tecnológicos de la Universidad de Nariño.

INTRODUCCIÓN

La Unidad de Informática y Telecomunicaciones (UIT) de la Universidad de Nariño, gestiona diferentes recursos informáticos para las prácticas académicas y de investigación de la comunidad universitaria. Dentro de los servicios que ofrece la UIT está el préstamo de las salas de cómputo, a las cuales debe ejecutar el mantenimiento preventivo y correctivo con el fin de brindar a los usuarios equipos que contribuyan en el desarrollo de sus actividades académicas, también tiene como función brindar el servicio de asistencias técnicas en temas de informática y telecomunicaciones, entre otros.

Mediante el desarrollo de este proyecto se ha creado un software que apoye la gestión de algunos procesos en la UIT, dentro de los cuales están: horarios de monitores técnicos y medio tiempo, horarios salas de cómputo para prácticas académicas y acceso a internet, reserva de equipos de cómputo por estudiantes, docentes o funcionarios, préstamos de equipos informáticos para la comunidad universitaria. De igual forma, dentro de la aplicación se encuentra contemplado el desarrollo de un módulo que administre el mantenimiento preventivo y correctivo de los equipos de cómputo de las unidades académicas y administrativas, entre otras actividades más que se verán integradas en este sistema de información.

En este documento, se presenta de una manera organizada la explicación de todo lo pertinente al desarrollo del sistema: el tema, los objetivos, la línea de investigación, los recursos, el cronograma de actividades, ente otras tareas más que se llevaron a cabo durante el transcurrir del proyecto.

1. MARCO TEÓRICO

1.1 UIT

La Unidad de Informática y Telecomunicaciones o también conocida como Aula de Informática de la Universidad de Nariño es un organismo que depende de la Vicerrectoría Académica, constituido como un laboratorio donde se desarrollan las actividades académicas de la Universidad que tienen que ver con la informática, la ingeniería de sistemas y las telecomunicaciones. Contando con un equipo de trabajo que está facultado para prestar una serie de servicios relacionados con la informática, la ingeniería de sistemas y las telecomunicaciones, en pro de colaborar con el desarrollo de las actividades de la Universidad de Nariño.

1.1.1 Áreas.

Figura 1. Áreas UIT



Fuente: <http://ainfo.udenar.edu.co/wp-content/uploads/2012/03/direcci%C3%B3n.png>

1.1.2 Servicios. Todos los servicios prestados por la UIT están relacionados con los recursos tecnológicos que posee la Universidad, cada uno es administrado por una de las áreas que pertenecen a esta dependencia (ver figura 1), de esta manera, el personal más especializado se encarga de manejar los servicios prestados que se dejan a su cargo, entre los servicios más importantes están:

- Administración del servicio de correo, tanto institucional como estudiantil.
- Administración de los sistemas de información que se encuentran instalados en los equipos que están ubicados en la granja de servidores, cuya gestión está a cargo de la UIT.
- El préstamo de los recursos tecnológicos (Aulas, Equipos Computo, Video beams, Sala de Proyecciones, etc.) disponibles al público y que están a cargo de la UIT.
- Gestionar y administrar el servicio de internet que se proporciona tanto a Docentes, Funcionarios y Estudiantes que pertenecen a la institución, así como también gestionar los recursos de red físicos (LAN) que posee la UDENAR.
- Brindar acceso a la red RENATA y a todos los servicios que esta presta a la comunidad estudiantil.
- Creación y administración algunos sitios web necesarios para los procesos e información de la comunidad universitaria, entre ellos, la página principal de la Universidad.
- Mantenimiento de los equipos de cómputo existentes dentro de las instalaciones, para su buen funcionamiento.
- Brindar los recursos necesarios, así como también proporcionar el personal idóneo para orientar los módulos de lenguaje y herramientas informáticas, los cuales están a cargo de la UIT.
- Brindar acceso adecuado a la red eléctrica de la institución.

1.2 SISTEMA GESTOR DE BASES DE DATOS MS SQLSERVER 2012

Es un sistema para la gestión de bases de datos, que ofrece un almacén de datos completo, eficaz y confiable para sitios web ligeros y aplicaciones de escritorio, creado por Microsoft y basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

1.2.1 Características.

- Soporta transacciones.
- Soporte de procedimientos almacenados.

También incluye un entorno gráfico de administración, lo que facilita el uso de comandos biom y DML gráficamente.

Es posible trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la misma.

Permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005, 2008 y 2012 pasa a ser el SQL Express *Edición*, que se distribuye en forma *gratuita*.

Es común desarrollar completos proyectos complementando *Microsoft SQL Server* y *Microsoft Access* a través de los llamados **ADP** (Access Data Project). De esta forma se completa la base de datos (*Microsoft SQL Server*), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

En el manejo de SQL mediante líneas de comando se utiliza el SQL CMD, o SQL, o *Power Shell*.

Para el desarrollo de aplicaciones más complejas (tres o más capas), *Microsoft SQL Server* incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos.¹

1.2.2 T-SQL. Permite interactuar con el servidor de la manera más efectiva. Por medio de este, es posible ejecutar las operaciones más importantes e indispensables en SQL Server, haciendo viable inclusive la creación y modificación de schemes de la base de datos, la introducción y edición de los datos en la base de datos, así como la administración del servidor como tal. Esto se realiza mediante el envío de sentencias de T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente.

T-SQL agrega programación procedural, variables locales, funciones de soporte para procesamiento de *strings*, procesamiento de fechas, matemáticas, etc. y cambios a las sentencias *DELETE* y *UPDATE*, en otras palabras, es una mejora al Estándar SQL.

¹ MICROSOFT SQL SERVER. En: Wikipedia [En Línea]. Disponible en: <http://es.wikipedia.org/wiki/Microsoft_SQL_Server> (Citado en 04 de febrero de 2014)

1.3 IDE MICROSOFT VISUAL STUDIO 2012

Visual Studio es una compilación completa de herramientas y servicios que le facilita la creación de una gran variedad de aplicaciones, colocando a disposición un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, *Phyton*, Ruby, php; al igual que entornos de desarrollo web como ASP.NET MVC, Django, etc., a lo cual sumarle las nuevas capacidades online bajo Windows *Azure* en forma del editor Mónaco.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas (la xbox 360 y xbox one), etc.

1.3.1 Novedades de Ms Visual Studio 2013. Visual Studio 2013 incluye muchas mejoras en la interfaz de usuario basado en la retroalimentación de los clientes y el principio de diseño de la base de Microsoft de mantener el enfoque en el contenido para ofrecer una experiencia de usuario mejorada.

Las características de productividad como la finalización automática del soporte, la capacidad de mover las líneas de código o bloques mediante atajos de teclado y navegación en línea están incorporados. Además, Visual Studio 2013 incluye un nuevo *Team Explorer* con una página de inicio mejorada que proporciona una navegación más fácil hacer equipo funciones de colaboración, incluyendo la posibilidad de desacoplar los cambios pendientes y construye en una ventana separada.

Code Lens (Código Indicadores de Información) en Visual Studio 2013 utiliza la información de ambos, su metadatos del proyecto y *Team Foundation Server* 2013 para colocar decoradores en cada uno de los métodos en el código, estos muestran información como los cambios llevados a la creación o cambio reciente, quien era la última persona que trabajo en el código, las pruebas de lo que cubre el método de la unidad (incluyendo el último estado pasa / falla), y existen referencias de los códigos. También puede saltar rápidamente a los datos, tal como un conjunto de cambios asociado o archivo de código hace referencia con sólo un clic.

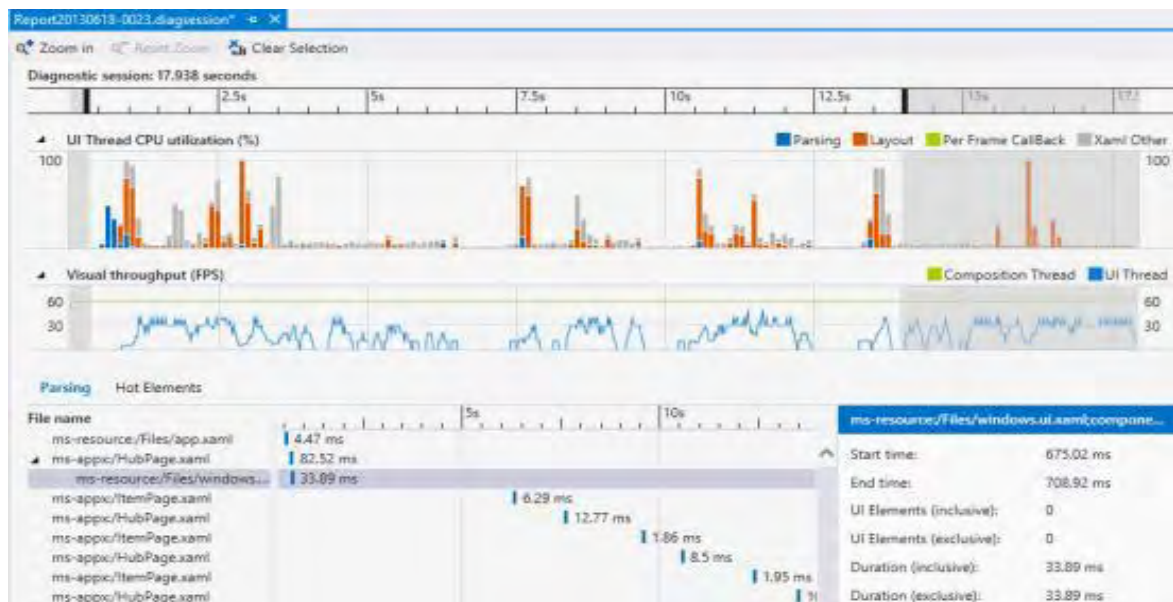
Visual Studio 2013 presenta una experiencia IDE conectado, lo que le permite empezar a trabajar con Visual Studio 2013 con mayor rapidez. Al iniciar Visual Studio 2013, por primera vez, se le pedirá que inicie sesión con su cuenta de Microsoft, que es posible que utilizar con sus suscripciones a MSDN o *Team Fundación Service*. Una vez conectado, su copia de Visual Studio 2013 se ha

registrado para su uso sin necesidad de ningún paso adicional. Visual Studio 2013 también se sincronizará automáticamente la configuración a través de sus dispositivos utilizando la experiencia IDE conectado. Los ajustes sincronizados incluyen categorías tales como atajos de teclado, la apariencia de Visual Studio (temas, fuentes, etc.) y más, a la vez que mantienen en control total el proyecto a través de las opciones de configuración sincronizada.

APOYO AL DESARROLLO DE WINDOWS 8.1 APP

Visual Studio 2013, proporciona el conjunto de herramientas ideal para construir aplicaciones modernas que aprovechan la nueva ola de innovaciones en la plataforma de Windows (Windows 8.1), mientras apoya a los dispositivos y servicios a través de todas las plataformas de Microsoft. El soporte para Windows Store de desarrollo de aplicaciones en Windows 8.1 incluye muchos cambios a las herramientas, controles y plantillas, compatibilidad con nuevas pruebas de IU codificadas para aplicaciones XAML, la interfaz de usuario de respuesta del analizador y generador de perfiles de consumo de energía para aplicaciones XAML y HTML, memoria mejorada de perfiles de herramientas para aplicaciones HTML, y una mejor integración con la tienda de Windows.

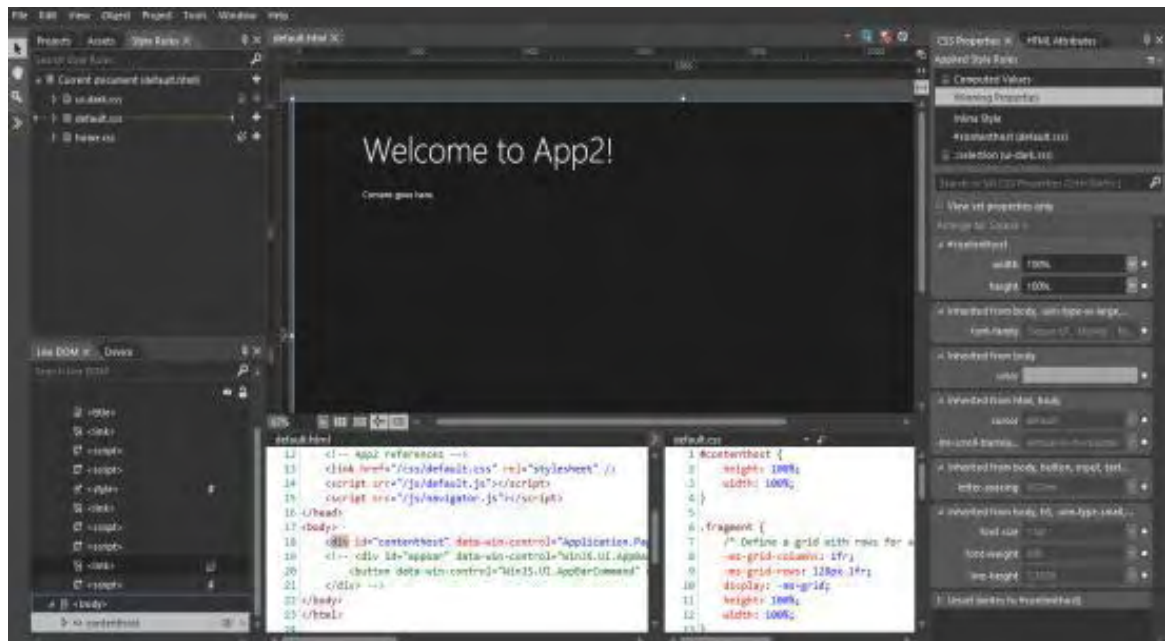
Figura 2. Windows Store



Blend de XAML se ha mejorado con nuevas capacidades para diseño de aplicaciones de Windows Store, tales como la adición de reglas y guías personalizadas para diseño de contenido más preciso, así como la mejora de las

capacidades de estilo. Mezcla para HTML también hereda estas mejoras y añade nuevas capacidades específicas para la construcción de ventanas de la tienda de aplicaciones con HTML, como la nueva línea de tiempo para la animación de los cambios en la CSS (ver figura 3).

Figura 3. Interfaz de Blend



La experiencia de codificación para Windows Store apps se ha mejorado en Visual Studio 2013. El editor XAML ahora proporciona avances en *Intelli Sense*, que ahora también está disponible para el enlace de datos, la unión de plantilla y recursos. También permite una transición más productiva entre el marcado diseño y código detrás, incluido el apoyo para la búsqueda de referencias, ir a la definición, y el cambio a la modelo de vista. Estos, junto con otras mejoras ayudarán a reducir el trabajo al crear aplicaciones de Windows Store.

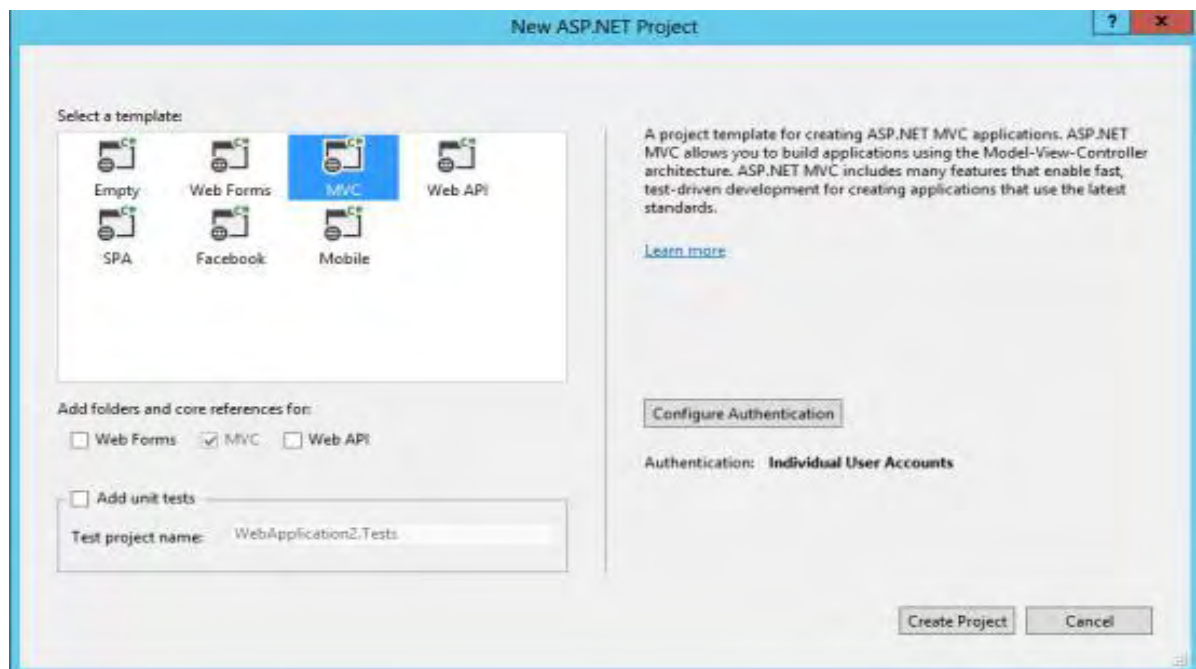
Conectar sus Windows Store Apps también es más fácil con la adición de *Windows Azure Mobile Services* que exponen su aplicación a una gran cantidad de escenarios tales como habilitar fácilmente notificaciones *Push* y el almacenamiento y manipulación de datos en la nube.

LOS AVANCES DE DESARROLLO WEB

Creación de sitios web o servicios en la plataforma de Microsoft le ofrece muchas opciones, incluyendo ASP.NET *Web Forms*, ASP.NET MVC, WCF o servicios

Web API, etc. En el pasado, el trabajo con cada uno de estos enfoques en Visual Studio significaba trabajar con tipos de proyectos separados y herramientas aisladas a las capacidades de ese proyecto específico. El de ASP.NET Visio unifica su experiencia en proyectos web en Visual Studio 2013 para que pueda crear aplicaciones web ASP.NET usando su preferencia de marcos de componentes ASP.NET en un único proyecto. Esta nueva experiencia unificada incluye la posibilidad de crear fácilmente aplicaciones híbridas que incluyen versiones mejoradas de ASP.NET *Web Forms*, MVC o API Web, todo en un solo proyecto. Ahora se puede mezclar y combinar las herramientas adecuadas para el trabajo dentro de los proyectos web, ofreciendo una mayor flexibilidad y productividad en el desarrollo de la web.

Figura 4. Tipos de Proyectos



En el diseño de páginas web, o implementar los diseños de otros, es probable que se gaste una parte importante de tiempo retocando HTML y CSS, mientras que las ventanas del navegador se refrescan una y otra vez para conseguir la experiencia exacta que desea crear. Esto se complica aún más por la diversidad de navegadores web disponibles. Visual Studio 2013 elimina este ir y venir a través del *Browse Link*, que conecta Visual Studio a uno o más navegadores modernos en una experiencia de desarrollo interactivo para mantenerlos en sincronía con los cambios de marcado sin necesidad de refrescar manualmente el navegador para ver los cambios que se están haciendo.

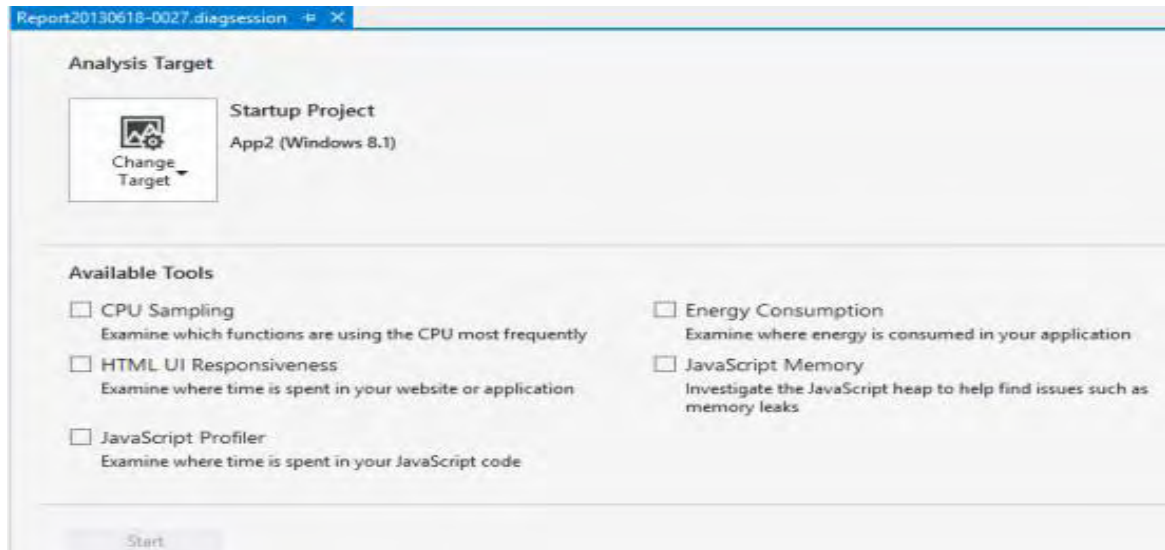
CREAR MODERN BUSINESS APPS PARA OFFICE 365

Con Visual Studio 2013, puede crear y gestionar Business Apps modernas que amplían la experiencia de Office 365 para ayudar a que las personas interactúen con los procesos de negocio, artefactos y otros sistemas. Office 365 Cloud Business Apps se ejecuta en la nube, aprovechando las ricas capacidades de la plataforma expuestas por Windows Azure y Office 365. Como las aplicaciones basadas en la nube, que están disponibles para una gran variedad de dispositivos para agregar datos y servicios de dentro y fuera de la empresa, e integrar las identidades de usuario y gráficos sociales. Estas aplicaciones se integran con las capacidades de gestión del ciclo de vida de aplicaciones de Visual Studio, uniendo los mundos del desarrollador de aplicaciones de negocios con las operaciones de TI.

MEJORAS DE DEPURACIÓN Y OPTIMIZACIÓN

El nuevo centro de Rendimiento y Diagnóstico en Visual Studio 2013 ofrece una experiencia unificada para diversas herramientas disponibles en Visual Studio. El centro también hace que sea fácil de identificar el rendimiento y herramientas de diagnóstico que trabajan con su proyecto y le permite comenzar rápidamente con ellos para mejorar su aplicación.

Figura 5. Ejemplo de Analizador de Rendimiento



El analizador de volcado de memoria en Visual Studio 2013 hace que sea fácil diagnosticar los problemas de memoria en entornos de producción a través del análisis de los volcados de memoria dinámica de .NET desde Visual Studio.

Utilizando la nueva capacidad de análisis de la memoria, se puede solucionar las pérdidas de memoria y el uso de memoria ineficiente al ver el tamaño y el recuento, las mayores instancias de objetos, y lo que es la celebración de su vida en la memoria. También puede comparar dos volcados de memoria para tener una idea de cómo el uso de memoria está cambiando con el tiempo.

Durante la depuración de 64-bit .NET 4.5.1 *applications* en *Visual Studio 2013*, se puede "editar-y-continuar" tal como lo hace con aplicaciones administradas de 32 bits. Además, la depuración asíncrono se mejora a través de Visual Basic, Visual C #, Visual C++ y Java Script en Visual Studio 2013. Si ejecuta en Windows 8.1, la ventana de la pila de llamadas ahora incluye la pila de llamadas lógicas, si se toca un *break point* o *excepción* en un método asíncronico. Más allá del apoyo diagnóstico, NET 4.5.1 incluye otras mejoras, como el soporte para la compactación por demanda de objetos grandes de la GC's, y más rápido inicio de aplicaciones cuando se ejecutan en máquinas *Multicore*.

En Visual Studio Ultimate 2013, puede diagnosticarse errores y problemas de rendimiento en aplicaciones monitorizadas por *System Center 2012 R2 Operations Manager*. Puede importar eventos de excepción y de rendimiento como *Intelli Tracelog* (.ITrace) y utilizar *Intelli Trace* de Visual Studio para desplazarse por la ejecución de código para esos eventos.

Además de las características que mejoran soporte para C++11, como constructores de delegados, *stringliterals* puros, operadores de conversión explícita y *variadic templates* junto a "*just my code*" *debugging* en Visual Studio 2013 hace que trabajar con C++ sea mejor. La capacidad de filtrar la pila de llamadas hasta sólo el código que escribiste al depurar ha existido por mucho tiempo para lenguajes administrados y ahora está disponible para C++. PGO (*Profile Guided Optimization*) es una optimización de compilador de ejecución de Visual C++ que aprovecha los datos de perfiles recopilados de funcionamiento importantes o el centro de rendimiento en escenarios de usuario, para construir una versión optimizada de una aplicación. Las herramientas de generación de perfiles de Visual Studio 2013 dan una idea de las decisiones de optimización de PGO, que ayuda a entender lo bien que los datos de perfil se corresponde con la actuación real del mundo, mostrando cómo los perfiles con rutas de código más calientes (escenarios de usuarios comunes) y las rutas de código más frías (menos común escenarios de usuario) de una aplicación se corresponden con lo que el código se optimizó mediante PGO y también cómo se hizo (tamaño o velocidad).

CAPACIDADES AMPLIADAS DE ALM

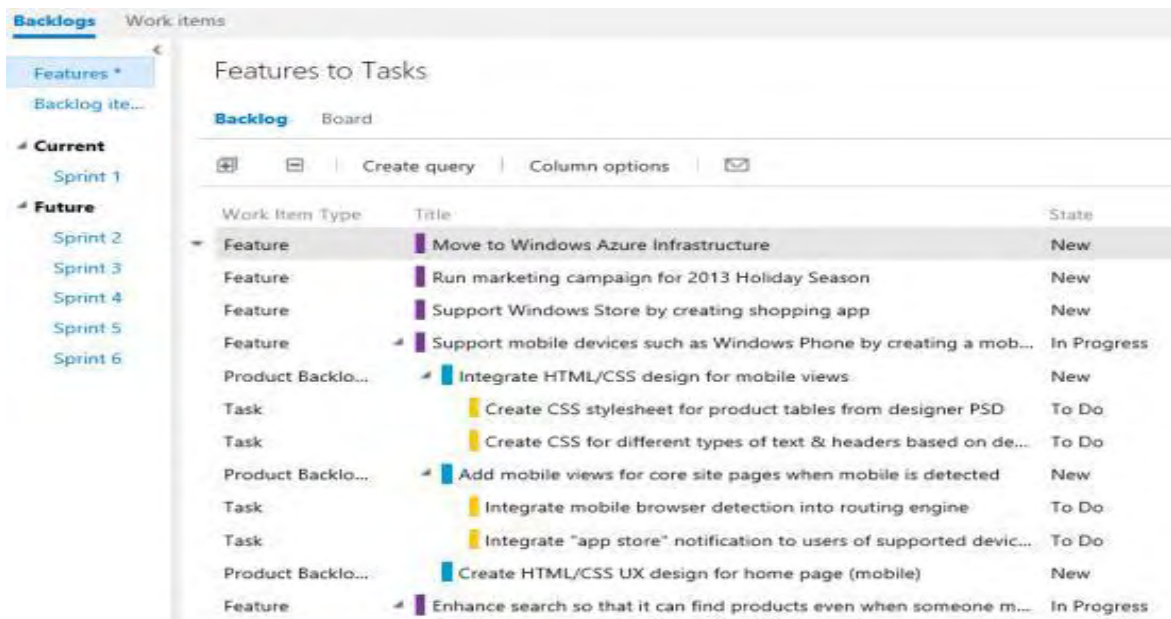
Visual Studio 2013 incorpora una oleada de nuevas características de *Application Lifecycle Management* (ALM) que le ayudan a sus equipos a ser más productivos y colaboran con soporte mejorado para las prácticas de desarrollo ágil que puede adoptar gradualmente para adaptarse a las necesidades de su organización.

Características tales como la gestión ágil de la cartera, salas de equipos, comentarios de código basado en la web sobre los *changesets/commits*, y caso de prueba basada en la web, mejoras en la gestión, son parte de una solución integrada para ayudar a su equipo de colaborar mejor y aumentar la productividad.

El núcleo de las capacidades de ALM de Visual Studio proporciona *Team Foundation Server*, que puede ser instalado en un servidor en el mismo recinto, o *Team Foundation Service*, que está alojado en la nube. Así que si el equipo prefiere el mayor control y personalización de una instalación o la accesibilidad sin necesidad de mantenimiento de la nube, hay una opción que se ajuste a las necesidades de su equipo.

Con Agile Portfolio Management puede administrar cargas de trabajo grandes y complejas de una forma ágil. Se puede definir una jerarquía de los retrasos para comprender el alcance del trabajo a través de varios equipos y ver cómo ese trabajo se desencadena en iniciativas más amplias.

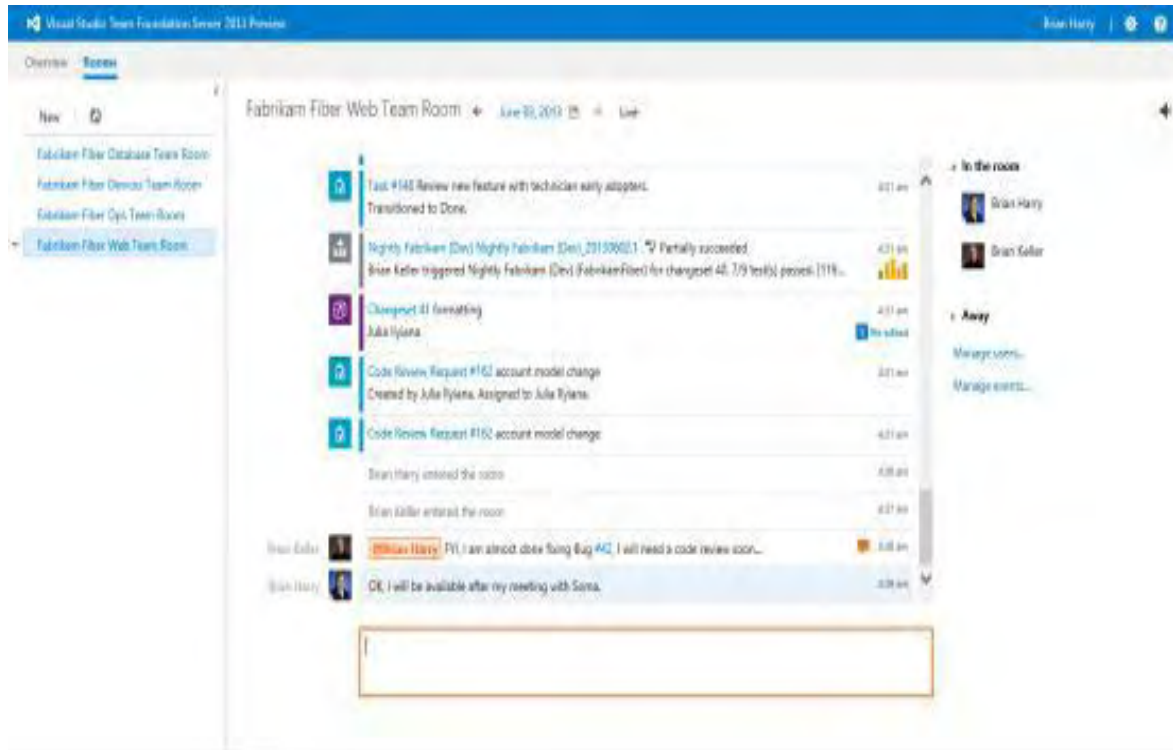
Figura 6. Agile Portfolio Management



Team rooms ofrecen un espacio para el fomento y la captura de la comunicación entre los miembros del equipo, tanto cerca como lejos. En un equipo de la habitación se puede hablar de trabajos en curso, hacer preguntas, estatus social, y aclarar las cuestiones que puedan surgir. Su habitación es su equipo de proyectos de equipo *Feed* de actividad para mantener al equipo informado de código de *check-in's*, su nivel de versión asignación de trabajo, y más. Mediante el uso de la

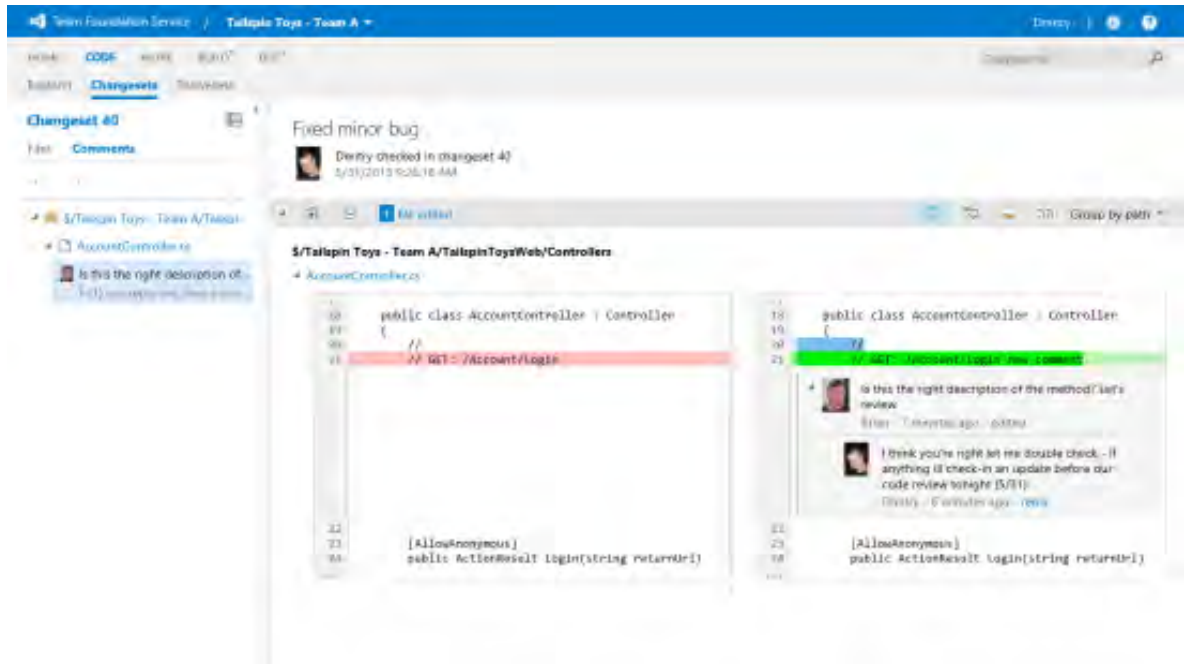
sala de equipo en vez de correo electrónico, recibirá automáticamente una pista de auditoría de las conversaciones y decisiones que puede consultar en cualquier momento y en esta categoría. Un equipo de la habitación se crea de forma predeterminada para cada proyecto, y se pueden crear salas adicionales para apoyar las discusiones ad-hoc y la colaboración entre equipos.

Figura 7. Team Rooms



Team Foundation Server 2013 ahora es compatible con los repositorios Git. Además, ahora se puede utilizar la versión del Control Explorer basado en la web en Visual Studio 2013 para crear o responder a los comentarios sobre los conjuntos de cambios aplazados, o *Gitcommits* para entablar conversaciones con su equipo sobre cambios en el código fuente. También se puede encontrar comentarios fácilmente usando el filtro Versión Explorer.

Figura 8. Team Foundation Server

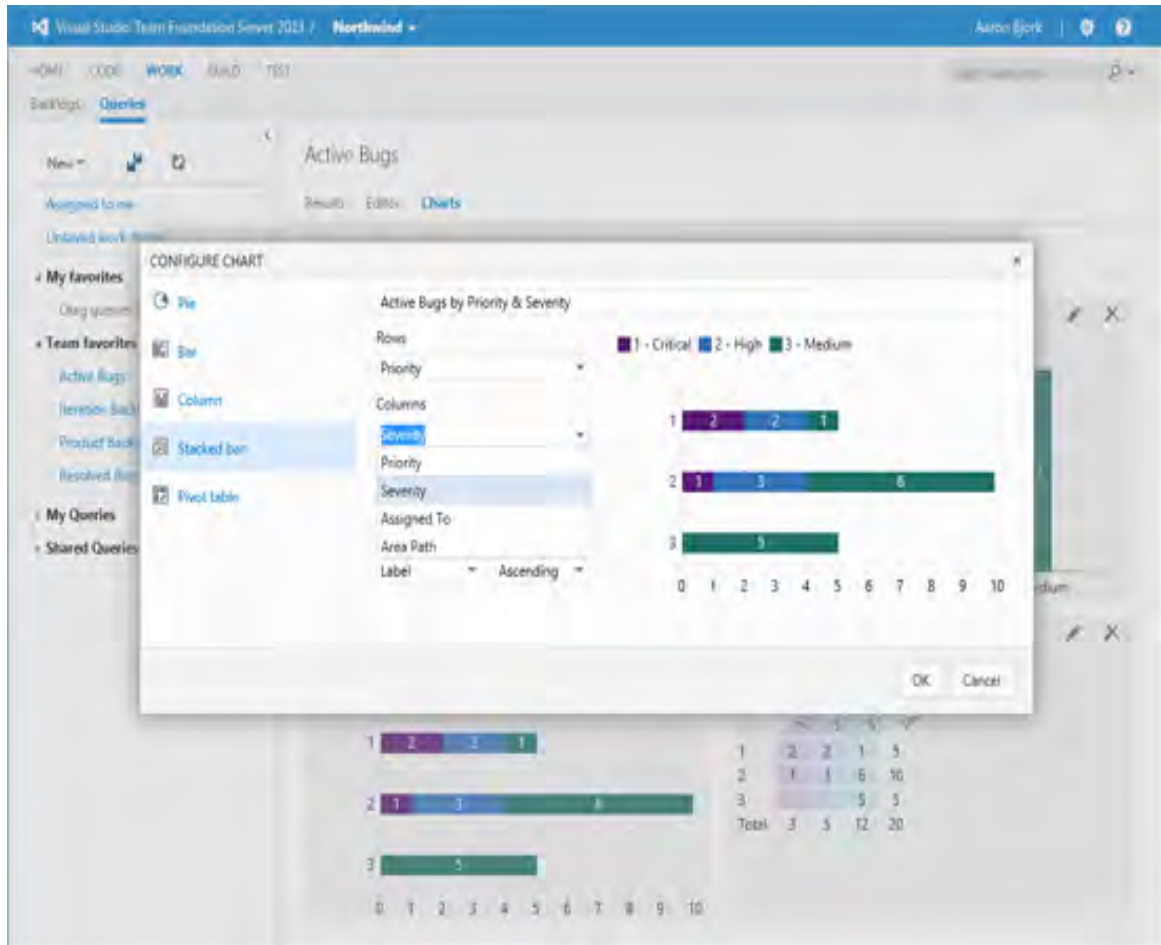


Con soporte para las pruebas de carga basado en la nube en *Team Foundation Service*, puede generar pruebas a escala en segundos que utilizan los recursos elásticos de *Windows Azure* para la ejecución de las peticiones de los miles de usuarios virtuales, que le proporciona una idea de los cuellos de botella de rendimiento de sus aplicaciones web y servicios sin la molestia de la gestión de hardware o un laboratorio virtual.

Las mejoras en la basada en la web de gestión de casos de prueba permiten crear o administrar planes y suites de prueba directamente desde la interfaz web. También puede ejecutar pruebas de sistemas operativos que no sean *Windows* utilizando cualquier navegador web moderno, permitiendo a sus probadores mantener la productividad de cualquier entorno de trabajo.

También es nuevo en *Team Foundation Server 2013* el elemento de trabajo de gráficos. Este crea rápidamente una variedad de gráficos para visualizar datos basados de sus consultas de elementos de trabajo, como los bugs, las historias de usuario y tareas.

Figura 9. Gráficos con Team Foundation Server



1.4 NET FRAMEWORK

Es una solución a toda la problemática en torno al desarrollo de aplicaciones, brinda grandes beneficios no solamente al desarrollador, sino también al proceso de desarrollo. En primer lugar .NET permite trabajar con código ya existente, podemos hacer uso de los componentes COM, e incluso, si lo necesitáramos usar el API de Windows.

Cuando el programa .NET está listo es mucho más fácil de instalar en la computadora de los clientes, que las aplicaciones tradicionales ya que se tiene una integración fuerte entre los lenguajes.²

² LANDA, Nicolás. C#. Guía total del programador. Buenos Aires: Fox Andina – Lomas de Zamora, 2010. p. 15.

Figura 10. Estructura de .NET Framework



Fuente: http://upload.wikimedia.org/wikipedia/commons/thumb/3/30/Diagrama_.NET.jpg/756px-Diagrama_.NET.jpg

1.4.1 Objetivos del .Net Framework. El .NET Framework fue diseñado para satisfacer los siguientes objetivos:

- **Basado en Estándares y Prácticas Web.** El NET Framework soporta totalmente las tecnologías existentes de Internet, incluyendo *Hypertext Markup Language* (HTML), XML, SOAP, *Extensible Stylesheet Language for Transformations* (XSLT), Xpath y otros estándares Web. El .NET Framework promueve la creación de servicios Web débilmente conectados y sin estado.
- **Extensible.** La jerarquía del .NET Framework no queda oculta al desarrollador. Se puede acceder y extender clases .NET (a menos que estén selladas) utilizando herencia. También es posible implementar herencia *multi-lenguaje*.
- **Fácil de usar por los Desarrolladores.** En el .NET Framework, el código está organizado en espacios de nombres jerárquicos y clases. El Framework proporciona un sistema de tipos común, denominado sistema de tipos unificado, que utiliza cualquier lenguaje compatible con .NET. En el sistema de tipos unificado, todo es un objeto. No existen tipos *variant*, sólo hay un tipo de cadena (*string*), y todos los datos de la cadena son Unicode.

- **Diseñado Utilizando Modelos de Aplicaciones Unificados.** La funcionalidad de una clase .NET está disponible desde cualquier modelo de programación o lenguaje compatible con .NET.

1.4.2 Novedades de .Net Framework 4.5.1.

- Recolección y compactación de elementos poco utilizados.
- Capacidad de recopilar información de diagnóstico para ayudar a los desarrolladores a mejorar el rendimiento de las aplicaciones de servidor y en la nube.
- Mejora del rendimiento, optimizando el procesamiento *multi-core* y disminuyendo el tiempo de inicio de las aplicaciones.
- La depuración de una aplicación se mejoró, y en visual studio 2013 fácilmente se podrá visualizar los tipos de datos y los valores devueltos por los métodos.
- Mayor compatibilidad con las excepciones de los componentes de Windows en tiempo de ejecución

Estas son solo algunas de las mejoras que han sido implementadas en el nuevo .Net Framework 4.5.1

1.5 PHP

PHP es un lenguaje de programación web del lado del servidor, su sigla significa *Hypertext Pre-processor* (aunque inicialmente era Personal Home Page). En sus inicios, PHP era muy sencillo, y era utilizado para crear contadores de visitas, páginas web simples, libros de visitas, etc. En la actualidad es un lenguaje muy robusto, con la capacidad de desplegar aplicaciones MVC e inclusive, hacer uso de *FRAMEWORKS*.

Es un lenguaje fácil de desplegar en casi todos los servidores web y casi en todos los sistemas operativos y lo mejor de todo, es gratis.

Permite conexión con muchos motores de Bases de datos, los cuales pueden ser SQL o NoSQL.

1.5.1 Características de PHP. Es un lenguaje de programación del lado del servidor, orientado al desarrollo de aplicaciones web, se ayuda del servidor web para interpretar el código PHP y generar la página resultante. Su sigla PHP Significa *Hypertext Pre-processor* (inicialmente PHP Tools, o, Personal Home Page Tools).

Entre sus principales características están:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de `phparrays`.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos. Incluso aplicaciones como Zendframework, empresa que desarrolla PHP, están totalmente desarrolladas mediante esta metodología.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.³

³ PHP. En: Wikipedia [En Línea]. Disponible en: <<http://es.wikipedia.org/wiki/PHP>> (Citado en 02 de Mayo de 2014)

2. METODOLOGÍA

Es un sistema de métodos que sirven como base de una disciplina para el desarrollo de una actividad, si se habla del proceso de desarrollo de software, se hablara de metodología de desarrollo de software.

De las metodologías de producción depende en gran parte la competitividad de una organización, ya que de esta dependen la calidad de los productos, las utilidades, el tiempo empleado en su elaboración y muchos factores más.

Estos beneficios/perjuicios aplican a cabalidad en una empresa de desarrollo de software, ya que estas también ofrecen un producto a sus clientes, y los ítems a tener en cuenta son generalmente los mismos.

2.1 METODOLOGÍAS DE DESARROLLÓ DE SOFTWARE

Las metodologías de desarrollo han sufrido grandes innovaciones desde los años 70, cuando se empezó a organizar los procesos y etapas para la producción de software, uno de los más aceptados por muchas de las organizaciones productoras de software fue el Modelo en Cascada, en el cual ya había una división de cada una de las etapas, más la gran desventaja que presenta esta propuesta, es que las fases no podían ser superpuestas. Generalmente, los pasos a seguir de esta metodología, son:

- Requisitos de sistema.
- Requisitos de software.
- Análisis.
- Diseño del programa.
- Codificación.
- Pruebas.
- Puesta en marcha.

Debido al estricto orden de las etapas, este enfoque presentaba una serie de desventajas, como las siguientes:

- Falta de flexibilidad.
- Es difícil predecir todos los problemas con anterioridad.
- Se pierde conocimiento importante entre etapas.
- Falta de cohesión en el equipo.
- Defectos que no son detectados hasta la etapa de pruebas, entre otras.

Para mejorar estos impases que presentaba el modelo en cascada, se definió el Modelo en iterativo e Incremental, el cual tuvo gran acogida durante mucho tiempo.

Se lo pensó de esta manera, para que el software generado por este proceso, fuera mucho más estable, ya que los cambios se van dando con pequeños pasos y por lo tanto, no son tan traumáticos, por el contrario, con cada modificación se enriquece el proyecto.

Otra de las ventajas que presenta este modelo, es la mayor participación por parte del usuario final del sistema, con lo cual la redefinición de requerimientos es constante.

Posteriormente parece el Modelo en Espiral, en el cual se tiene una serie de iteraciones, las cuales solo están definidas por la iteración anterior y el análisis de riesgo de la misma.

El Modelo Iterativo e Incremental tuvo gran aceptación en la década de los 90 con algunas de sus formas más funcionales como por ejemplo: el prototipado rápido, Desarrollo Rápido de Aplicaciones RAD y el más popular, el Proceso Unificado de Rational RUP.

2.2 EL NUEVO ENFOQUE

Debido a múltiples factores, que acontecen durante el transcurso de un proyecto de software, los cuales afectan mucho al proceso de desarrollo y sobre todo al producto final, era necesario definir nuevos lineamientos que dieran solución a todos estos inconvenientes presentados.

Antes de hablar sobre las soluciones, se debe hablar un poco sobre los problemas que se presentaban:

- **Requerimientos cambiantes:** entre la toma de requerimientos y la entrega del producto final, los requerimientos pueden cambiar, debido a cambios en la empresa, lo cual genera un producto que no se adecua a las necesidades manifestados por el cliente.
- **Los clientes dan requerimientos herrados:** los usuarios finales del sistema pueden dar información falsa o irrelevante e inclusive, inventar requerimientos, con esto, van a confundir al equipo de desarrollo, causar daños en el sistema a desarrollar y a retrasar la entrega del producto final.

Dado estos costos extras, es necesario cambiar el enfoque donde el cliente

percibe valor sólo al final del desarrollo, por uno donde el cliente reciba las funcionalidades de manera iterativa a medida que avanza el desarrollo del producto final.

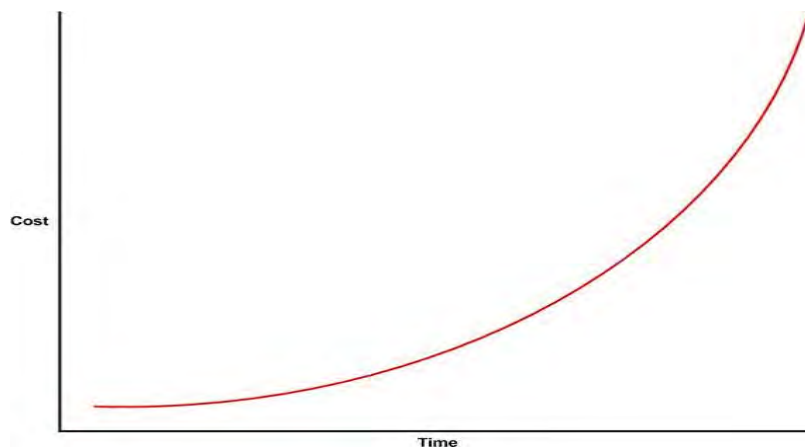
2.3 METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE

Las metodologías ágiles de desarrollo de software proponen como primer paso, priorizar los requerimientos, poniendo en primer nivel, las funcionalidades de mayor necesidad para el cliente, para que el proceso de desarrollo se lleve a cabo de manera iterativa e incremental.

Lo primordial en este tipo de metodologías, es mantener el diseño siempre adaptable a cambios y sobre todo, que estos cambios no causen dificultades en el proyecto; como ayuda para esta tarea, se tiene la curva de costo de cambios, la cual muestra gráficamente cuánto cuesta realizar cambios en el software de acuerdo al tiempo de desarrollo transcurrido. Esta imagen fue propuesta por Barry Boehm y presenta los costos tanto de trabajo de desarrollo, como costos económico. Esta imagen puede variar de una metodología a otra, y entre más se eleva la curva, mayor será el costo de su cambio.

Esta es una imagen clásica en las metodologías en Cascada o Iterativo e Incremental

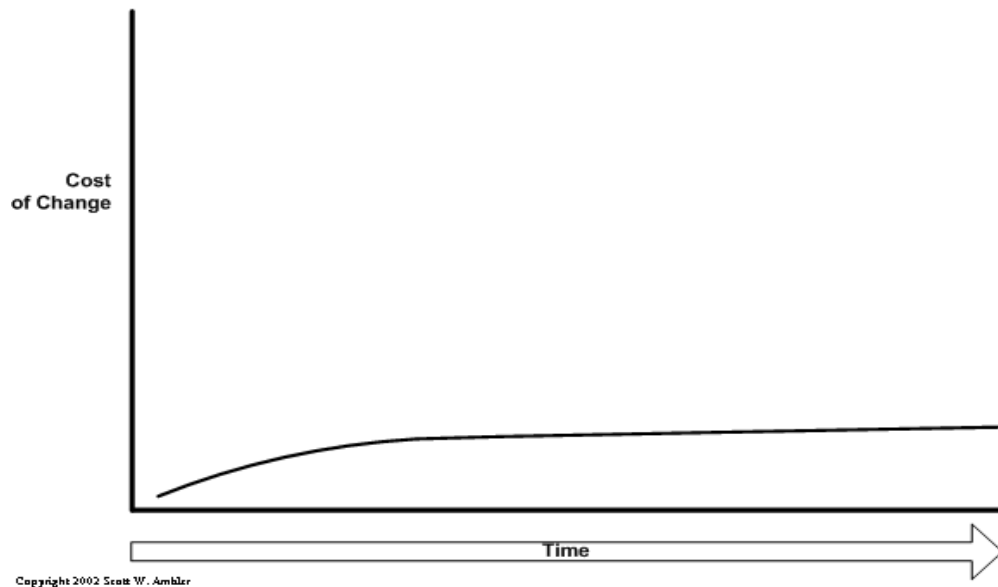
Figura 11. Grafica Costo-Tiempo Modelo en Cascada o Modelo Incremental



Fuente:http://1.bp.blogspot.com/_ETyQOZyYlpY/TQ_ah90vwpl/AAAAAAAAACQ/JrR9kOhrtcE/s1600/scrum+y+xp.png

Mientras que si se mira la misma gráfica para una metodología de desarrollo ágil, la curva que se obtendrá será un poco más plana, como la siguiente:

Figura 12. Gráfica Costo-Tiempo Metodologías Ágiles de Desarrollo



Fuente: http://1.bp.blogspot.com/_ETyQOZyYlpY/TQ_ah90vwpl/AAAAAAAAACQ/JrR9kOhrtcE/s1600/scrum+y+xp.png

Las metodologías ágiles, no se apartan mucho de las metodologías tradicionales, sobre todo del modelo evolutivo, pero lo que hacen es explotar las ventajas que estas proporcionan, así lo hace la programación extrema o XP como se la conoce comúnmente. Una metodología que tiene como premisa iniciar el desarrollo por los objetivos más valiosos, dejando a los de menor importancia como valor agregado. Pero sobre todo, lo que busca esta metodología es eliminar la pesada carga que significa el papeleo para los desarrolladores.

El *Manifiesto for Agile Software Development* (Manifiesto para el Desarrollo Ágil de Software), establecen los principios del movimiento ágil, y que expresados de una manera muy sencilla dicen lo siguiente:

- Personas y sus interacciones por sobre procesos y herramientas.
- Software funcional por sobre documentación exhaustiva.
- Colaboración del cliente por sobre la negociación con contrato.
- Responder al cambio por sobre seguir un plan.

2.3.1 Extreme Programming (Programación Extrema).

Básicamente, la programación extrema, busca dos objetivos claramente: hacer un software bien (con calidad) y de la forma más rápida posible. De hecho estos son los objetivos fundamentales de cualquier metodología aplicada al

desarrollo de software y a cualquier otra área en general. A pesar de esto, con las metodologías de desarrollo actuales, el 70% de los proyectos fracasan y aproximadamente, también, el 70% de los fallos no son debidos a cuestiones técnicas, son debidos a cambios en la gestión o problemas de comunicación. Con estos datos es lógico pensar en que las metodologías actuales no son lo suficientemente buenas, porque una tasa de éxito inferior a una tercera parte del total de proyectos no es algo deseable.

Una vez analizado el problema, podemos ver en XP la solución, o al menos un acercamiento. La programación extrema centra su atención en la producción de software con una fuerte arquitectura, intentando sacar productos al mercado rápidamente, con gran calidad y motivando al equipo de trabajo para seguir mejorando esta tendencia.⁴

En su artículo “*Is Design Dead?*” Martin Fowler⁵ expresa que los diseños de un software traen muchos problemas en la etapa de codificación ya que los diseñadores desconocen las herramientas de desarrollo, también están los ya mencionados requerimientos cambiantes, lo que eleva el nivel de entropía así como también pérdida de tiempo mientras se corrige el error en el diseño.

La programación extrema trata de solucionar estos inconvenientes de una manera más práctica, por lo cual se propone un diseño evolutivo y modificable, sobre un diseño planificado.

Las 2 premisas fundamentales de la Programación Extrema son: “haz las cosas lo más sencillas posibles, pero que funcionen” y “esto no lo vas a necesitar, por lo tanto no lo hagas”, lo que implica solo desarrollar lo necesario y de la mejor manera posible, esto se debe tener muy en cuenta, cuando se habla de partes reutilizables, ya que no se recomienda no desarrollarlas a la primera iteración en la que aparecen, si no se sabe cuántas veces serán utilizadas.

La XP busca el fácil mantenimiento del aplicativo, utilizando nuevas tecnologías, que estén a la vanguardia, con códigos simples, comentarios explicativos de los procedimientos que se realizan en cada sección del código, variables con nombres bastante informativos y muchas técnicas más. Esto es una gran mejora sobre otras metodologías, ya que ni con sus inmensas documentaciones es posible entender algunas partes del código que esta implementado.

Una de las características más importantes de esta metodología, es la realización constante de pruebas a cada sección de código desarrollado, para lo cual es importante saber identificar los puntos donde el software puede presentar bugs, y hacer los test correspondientes, con los datos indicados para la adecuada

⁴ CORTIZO PEREZ, José; EXPOSITO G., Diego; RUIZ L., Miguel. eXtreme Programing, 2012. p. 8.

⁵ FOWLER, Martin. *Is Design Dead*, Martin Fowler [En Línea]. Disponible en: <<http://martinfowler.com/articles/designDead.html>> (Citado en 22 de febrero de 2014).

verificación y su posterior corrección. Pero las pruebas no terminan cuando los errores de cada sección de código han sido rectificadas, ya que también es importante verificar la integración con otras secciones, por lo cual, el testeo será constante mientras el proyecto está en desarrollo.

La refactorización es la parte del mantenimiento del código que no arregla errores ni añade funcionalidad. El objetivo, por el contrario, es mejorar la facilidad de comprensión del código o cambiar su estructura y diseño y eliminar código muerto, para facilitar el mantenimiento en el futuro. Añadir nuevo comportamiento a un programa puede ser difícil con la estructura dada del programa, así que un desarrollador puede refactorizarlo primero para facilitar esta tarea y luego añadir el nuevo comportamiento.⁶

Lo anterior hace referencia a una de las tareas más importantes de la programación extrema, ya que al ser una metodología en la cual se le da prioridad al código, es necesario mantenerlo organizado y simplificado al máximo.

Ya hemos dado las bases que rigen a la programación extrema, en las etapas de desarrollo y/o codificación, pero hay algo que no podemos pasar por alto, ya que es uno de los principales pilares en la Programación Extrema y se encarga de establecer las bases definidas por el cliente, para pasarlas al desarrollador, esto se hace con las denominadas historias de usuario, estas son una forma de expresar un requerimiento de un usuario utilizando una o dos frases, estas van acompañadas de discusiones con el usuario y las respectivas pruebas.

Figura 13. Esquema de Desarrollo en XP



Fuente: <http://4.bp.blogspot.com/->

[KoB3BMXOMEY/T4AI8b2SrdI/AAAAAAAAAJ0/ARq3GhNGI50/s1600/agileflowchart.gif](http://4.bp.blogspot.com/-KoB3BMXOMEY/T4AI8b2SrdI/AAAAAAAAAJ0/ARq3GhNGI50/s1600/agileflowchart.gif)

⁶⁶ REFACTORIZACIÓN. En: Wikipedia [En Línea]. Disponible en: <http://es.wikipedia.org/wiki/Refactorizaci%C3%B3n> (Citado en 28 de febrero de 2014)

3. SISTEMA DE INFORMACIÓN PARA EL MANEJO DE RECURSOS DE LA UNIDAD DE INFORMÁTICA Y TELECOMUNICACIONES DE LA UNIVERSIDAD DE NARIÑO

3.1 FASE DE PLANEACIÓN

3.1.1 Historias de usuario.

Tabla 1. Acceso al Sistema

Historia de Usuario	
Numero: 1	Nombre: Acceso al sistema
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Alto (Alto / Medio / Bajo)	Puntos Reales: 4
<p>Descripción: Esta historia de usuario establece que los usuarios tendrán un sistema de autenticación para ingresar a las opciones del sistema, el cual debe ser adecuado para que diferencie entre los usuarios registrados en el sistema de información integrado de la UIT y los funcionarios que vayan a realizar o calificar solicitudes de asistencia. También debe haber un sistema de autenticación para los usuarios que vayan a realizar solicitudes de préstamo de equipos.</p> <p>Para estas opciones se tendrá en cuenta el nivel de seguridad adecuado para cada uno de los casos y que además deben estar ligados con algunos de los sistemas de autenticación que ya están implementados en la Institución.</p>	
<p>Observaciones: Previamente a los formularios de autenticación, debe existir un menú donde el usuario pueda ubicar la opción más adecuada para su necesidad.</p>	

Tabla 2. Usuarios y Roles

Historia de Usuario	
Numero: 1	Nombre: Usuarios Y Roles
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 4
Descripción: Los usuarios podrán ser registrados, editados y deshabilitados por otros usuarios que tengan permisos para hacerlo, ingresando la información necesaria para la adecuada identificación de cada uno, además también deben poder administrarse roles de manera dinámica, y los permisos que tendrá cada rol sobre los diferentes recursos.	
Observaciones: Los usuarios con rol de administrador tendrán acceso a todas las opciones del sistema de información.	

Tabla 3. Crear, Calificar y Revisar el Historial de Solicitudes

Historia de Usuario	
Numero: 1	Nombre: Crear, Calificar y Revisar el Historial de Solicitudes
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Alto (Alta / Media / Baja)	Puntos Estimados: 5
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 5
Descripción: Los funcionarios de la UDENAR deberán poder registrar sus solicitudes de asistencia a través del sistema, diligenciando la información adecuada de ubicación, descripción, solicitante y fecha, para que estas sean atendidas posteriormente por parte de los funcionarios de la UIT. Se podrá calificar las solicitudes realizadas que hayan sido dadas por terminadas. Se debe poder eliminar las asistencias realizadas y que aún no se hayan empezado a desarrollar. Así como también es conveniente la opción de poder mirar un historial de las últimas solicitudes de asistencia realizadas.	
Observaciones:	

Tabla 4. Reserva de Equipos

Historia de Usuario	
Numero: 1	Nombre: Reserva de Equipos
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 4
Descripción: Los equipos que los funcionarios de la UIT consideren que deben ser habilitados para préstamo al público, podrán ser reservados a través del Sistema de Información Integrado, siempre y cuando previamente el recurso no haya sido reservado o este en estado de préstamo.	
Observaciones: La reserva de equipos de cómputo habilitados para préstamos de franjas de internet, deberán hacerse teniendo en cuenta los horarios de clase de las aulas de informática.	

Tabla 5. Inventario de Equipos

Historia de Usuario	
Numero: 1	Nombre: Inventario de Equipos
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Alto (Alto / Medio / Bajo)	Puntos Reales: 5
Descripción: El sistema facilitará el control de inventarios de equipos de cómputo y equipos similares que pertenecen a la universidad y que estén a cargo de la UIT; Este inventario deberá ser fácil de actualizar. Deberá tener mecanismos de control sobre la redundancia de información.	
Observaciones: También debe poder darse de baja los equipos que ya están en desuso.	

Tabla 6. Mantenimientos

Historia de Usuario	
Numero: 1	Nombre: Mantenimientos
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 4
<p>Descripción: El sistema permitirá llevar un historial de mantenimientos realizados a los equipos que estén registrados en la base de datos, clasificarlos si son de tipo preventivo o correctivo, así como una sub-clasificación del tipo de problema a solucionar, la fecha y también quien realizo dicho procedimiento.</p>	
<p>Observaciones: Si los procedimientos fueron realizados en cumplimiento de alguna asistencia para corregir problemas de funcionamiento a los equipos de alguno los funcionarios, deberá hacerse la respectiva relación.</p>	

Tabla 7. Historia de Usuario Solicitudes

Historia de Usuario	
Numero: 1	Nombre: Solicitudes
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Alto (Alta / Media / Baja)	Puntos Estimados: 5
Riesgo en Desarrollo: Alto (Alto / Medio / Bajo)	Puntos Reales: 5
<p>Descripción: Las solicitudes serán atendidas por el personal de la UIT que tenga autorización para este proceso, y luego serán asignadas o auto-asignadas a quien se crea conveniente (Monitores). Los monitores podrán revisar las solicitudes que les han sido asignadas y empezar a trabajar en las mismas, realizando procedimientos de mantenimiento a equipos. Una vez terminados los procedimientos, los usuarios podrán dar por terminada la asistencia.</p>	
Observaciones:	

Tabla 8. Horarios Monitores

Historia de Usuario	
Numero: 1	Nombre: Horarios Monitores
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Bajo (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 4
<p>Descripción: Se podrán crear los horarios necesarios para controlar la asistencia de los monitores a las diferentes jornadas de trabajo, teniendo en cuenta el tipo de monitoria y también imposibilitando que el monitor se ubique en 2 horarios al mismo tiempo. Se deberán poder eliminar los horarios que ya no estén vigentes.</p>	
Observaciones: Se deberán poder imprimir la versión final de cada horario	

Tabla 9. Aulas

Historia de Usuario	
Numero: 1	Nombre: Aulas
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 4
<p>Descripción: Se tendrá un registro de las aulas de informática que existen en la Universidad, así como los equipos que están ubicados en cada una de ellas. Se deberá poder controlar los horarios de las clases que se dictan en las Aulas de informática, de acuerdo a los requerimientos de cada programa o dependencia de la institución. Esto con el fin de que cuando los equipos estén libres, puedan ser usados para el acceso a internet por parte del personal que así lo requiera.</p>	
Observaciones: Los horarios de cada aula deben poder imprimirse.	

Tabla 10. Préstamos

Historia de Usuario	
Numero: 1	Nombre: Préstamos
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Bajo (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 4
Descripción: Los equipos que estén habilitados para préstamo y que sean solicitados por el personal perteneciente a la Universidad ya sea por internet o presencialmente, se deberán poder prestar, siempre y cuando se haya diligenciando los datos de la persona que los solicite.	
Observaciones: Deberá haber alguna manera de autenticar a la persona solicitante.	

Tabla 11. Reportes

Historia de Usuario	
Numero: 1	Nombre: Reportes
Usuario: ALBEIRO N. CHAMORRO P	
Modificación de Historia No.: N/A	Iteración Asignada: 1
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 4
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales: 4
Descripción: El sistema deberá generar los reportes que sean especificados y/o solicitados por los directivos tanto de la UIT, como de la Institución. Algunos reportes deberán tener la opción de filtrar los resultados, para así obtener la información más adecuada y conveniente para los funcionarios de la UIT	
Observaciones: Deben ser en un formato no editable.	

3.2 PLAN DE ENTREGAS

El plan de entregas del proyecto se ha programado según las historias de usuario, por lo cual se estableció el inicio y finalización de cada historia. A continuación se indica el cronograma de actividades que se tuvo en cuenta para la implementación de las historias de usuario:

Tabla 12. Plan de Entregas

No	Historia	Fecha de Inicio	Fecha de Fin	Semanas estimadas
1	ACCESO AL SISTEMA	03-02-2013	20-04-2013	10
2	USUARIOS Y ROLES	21-04-2013	01-06-2013	6
3	CREAR SOLICITUDES	02-06-2013	20-07-2013	7
4	RESERVA DE EQUIPOS	21-07-2013	14-09-2013	8
5	INVENTARIO DE EQUIPOS	15-09-2013	17-11-2013	9
6	MANTENIMIENTOS	18-11-2013	19-01-2014	8
7	SOLICITUDES	20-01-2014	23-02-2014	5
8	HORARIOS MONITORES	24-02-2014	23-03-2014	4
9	AULAS	24-03-2014	27-04-2014	5
10	PRESTAMOS	28-04-2014	25-05-2014	4
11	REPORTES	26-05-2014	08-06-2014	2

3.3 FASE DE DISEÑO

Metáfora del Sistema. Aquí se registran las especificaciones que se les plantean a los usuarios finales del sistema, con los cuales contará la aplicación.

- La solución será una aplicación web, la cual estará instalada en uno de los servidores de la universidad que esté a cargo de la UIT, y con sus respectivas conexiones y configuraciones de red, además tendrá su propio sub-domino institucional.
- La base de datos solo será modificada de manera directa, por personal autorizado de la UIT, el resto de usuarios registrados solo tendrán acceso a los recursos que les sea permitidos para su rol, el público en general podrá hacer solicitudes y/o reservar equipos.
- Los datos donde pueda generarse redundancia de información, serán controlados formalizando la base de datos hasta tercera forma normal, o desde la aplicación, usando controles que permitan selección de opciones preestablecidas.
- Donde haya que hacer búsquedas de equipos, estas se harán por diferentes ítems, los cuales pueden ser uno o varios e inclusive ninguno.

En la solución que se desea implementar se ha visto la necesidad de estructurarla en los siguientes módulos:

3.3.1 Módulo de usuarios.

- La aplicación web se deberá configurar para que funcione con un sistema de autenticación existente en la UDENAR.

- Los usuarios que pasen el filtro de autenticación de la Universidad, serán buscados en el sistema de información, si se encuentran en la base de datos, serán re-direccionados a la página principal del sistema de información integrado, los que por el contrario, no se encuentren registrados, serán re-direccionados a la sección de solicitudes de asistencia.
- Los usuarios administradores así como también los que tengan los permisos necesarios, podrán crear, editar o deshabilitar usuarios de la aplicación.
- Los usuarios administradores así como también los que tengan los permisos necesarios, podrán crear y eliminar roles, así como también editar los permisos que tendrá cada uno de estos sobre los diferentes recursos o servicios de la aplicación.

3.3.2 Módulo de solicitudes.

- Las solicitudes podrán ser creadas por usuarios externos al sistema, o por funcionarios de la UIT.
- Se registrará el nombre del solicitante, la descripción, el medio por el cual se realizó la solicitud, la fecha y la hora en la cual se creó.
- Una vez terminada la asistencia, el usuario deberá registrar una calificación de la calidad del servicio prestado, este procedimiento se llevara a cabo por cada solicitud que haya realizado. Si la calificación de la asistencia es negativa, el usuario deberá registrar obligatoriamente una justificación del ¿por qué? de esta calificación.
- Si el usuario que desea pedir una asistencia, tiene solicitudes anteriores pendientes de calificar, deberá registrar primero una valoración de la asistencia finalizada para poder realizar nuevas solicitudes.
- El usuario tendrá una opción para eliminar las solicitudes que por alguna razón ya no sea necesario darles solución y que aún no hayan sido empezadas a desarrollarse.
- Habrá una opción donde el usuario pueda revisar el historial de las últimas solicitudes realizadas.
- Existirá una opción donde se podrán revisar las solicitudes atendidas por los funcionarios de la UIT en un intervalo de fechas y también los mantenimientos vinculados a cada una de estas.
- Una vez realizada la solicitud, será asignada o auto asignada a uno de los monitores de la UIT, para que sea atendida lo más pronto posible.
- El monitor podrá revisar las solicitudes que le han sido asignadas, y podrá empezar a desarrollar la que el seleccione.
- Las solicitudes se vinculan a un equipo del inventario, a través de procedimientos de mantenimiento los cuales también pueden incluir cambios de hardware.
- Los procedimientos de mantenimiento deberán registrar como mínimo la siguiente información: el número del equipo al cual se le realizo el procedimiento,

el tipo de mantenimiento (preventivo, correctivo), una clasificación del tipo de problema, la fecha y hora, quien realizo dicho procedimiento y su respectiva descripción, también se especificara la solicitud de asistencia por la cual se realizó el trabajo.

- Cuando el problema haya sido solucionado, el funcionario de la UIT dará por terminada la asistencia, con lo cual esta quedara pendiente de recibir su correspondiente calificación por parte del solicitante.

3.3.3 Módulo de inventario.

- Se podrá registrar las características de los equipos que se encuentren a cargo de la UIT, los campos que sean generales para todos los equipos, deberán ser obligatorios. También debe haber la posibilidad de crear lotes de equipos, para facilitar el trabajo cuando las características de hardware sean similares.
- Se podrán editar las características de los equipos que estén a cargo de la UIT.
- Se podrán dar de baja los equipos que estén demasiado averiados, o que estén obsoletos.
- Se podrán agregar mantenimientos a equipos que así lo requieran, pero estos procedimientos no estarán ligados con alguna asistencia.
- A los equipos de cómputo se les podrá agregar o quitar dispositivos adicionales.
- Cuando un equipo requiera más de un procedimiento para solucionar los inconvenientes, este será puesto en mantenimiento, hasta que su reparación finalice, o sea dado de baja si ese es el caso. Los equipos que pasen a mantenimiento, no estarán disponibles para el uso de los diferentes usuarios de la UDENAR.

3.4.4 Módulo de horarios de monitores.

- Se deberá permitir crear los horarios que sean necesarios para llevar un control adecuado de los monitores y sus jornadas laborales.
- Se deberá poder editar los horarios creados.
- Se deberá poder eliminar los horarios existentes.
- Los monitores no deberán aparecer en dos horarios el mismo día, y a la misma hora.
- Los horarios laborales de cada monitor no deberán exceder las horas labóreles establecidas por el tipo de monitoria que cada uno tenga (15 horas para monitoria técnica y 20 horas para monitoria medio tiempo).
- Habrá la posibilidad de imprimir los horarios que ya estén finalizados.

3.4.5 Módulo de aulas de informática.

- El sistema deberá permitir crear las aulas de informática que existen en la institución, y a las cuales se vincularán los equipos de cómputo que se ubican en cada una de ellas.
- Algunas aulas tendrán un monitor responsable, encargado de velar por el adecuado funcionamiento de la misma.
- Se podrá crear los horarios de clase que se tiene en cada una de las aulas, de acuerdo a los requerimientos que entregue cada programa o dependencia para la respectiva asignatura y teniendo en cuenta que no se presenten cruces de aulas o docentes.
- Las franjas que queden libres, serán utilizadas para el acceso a internet por parte del personal que así lo requiera.

3.4.6 Módulo de reservas y préstamos de equipos.

- Se permitirá hacer la reserva de equipos de cómputo para el acceso a internet en las aulas de informática existentes en campus universitario, esto se hará utilizando un método de verificación para validar que los estudiantes estén activos.
- Permitirá reservar otros equipos que también se encuentran habilitados para el préstamo a los docentes y funcionarios de la Universidad.
- La entrega de los equipos que están disponibles para préstamo a los diferentes usuarios de la Universidad, se deberá obtener la información necesaria que garantice la devolución de parte del solicitante. Una vez entregado el equipo, será puesto en estado de préstamo, con lo cual no estará disponible para otros usuarios.
- Cuando el equipo sea devuelto, se dará por terminado el préstamo, para poner el equipo nuevamente como disponible para los otros usuarios.

3.4.7 Módulo de reportes.

- El sistema permitirá generar los reportes necesarios para rendir cuentas ante los directivos de la UIT, como de la Universidad.
- Algunos reportes tendrán la opción de realizar filtros, para obtener solo los datos pertinentes.
- Los reportes serán en un formato no editable, para garantizar que no sean alterados.

3.5 DISEÑO DE LA BASE DE DATOS

Tabla 13. Tablas de la Base de Datos


NOMBRE	DETALLE	TIPO
Aulas	Almacena la información de las aulas que han sido creadas en el sistema de información, así como el monitor que es responsable de las mismas.	Tabla
Dependencias	En esta tabla se almacenan los nombres de las diferentes dependencias que existen en la universidad.	Tabla
Dispositivosequipo	Se almacena los datos de los dispositivos extras que posee cada uno de los equipos.	Tabla
Equipos	Almacena la información de los equipos en general, y las características específicas si se trata de computadores.	Tabla
Horariosaulas	Aquí se registra la información concerniente a los horarios de las aulas de informática	Tabla
Horariosm	Guarda los registros donde se establecen los horarios laborales de cada uno de los monitores que están habilitados.	Tabla
Mantenimientosequipos	Se almacenan el historial de los mantenimientos realizados a un equipo, con todos los datos necesarios.	Tabla
Permisos	En esta tabla se almacena los permisos que puede tener un rol sobre los diferentes recursos que existen en el sistema	Tabla
Prestamos	En esta tabla se almacenan los datos sobre los equipos de cómputo de las aulas de informática que se prestan a los diferentes usuarios por parte de la UIT.	Tabla
Prestamosotros	En esta tabla se almacenan los datos sobre los equipos (Videobeams, Micrófonos, Parlante, Cables, Etc.) que se prestan a los diferentes usuarios por parte de la UIT.	Tabla
Programas	En esta tabla se consignaran los diferentes programas que hacen uso de las aulas de informática que están a cargo de la UIT y a las cuales se les hace control de horarios.	Tabla
Recursos	Esta tabla almacena los recursos a los cuales los usuarios pueden acceder, siempre y cuando su Rol tenga permisos para hacerlo.	Tabla
Roles	Aquí se almacenan los diferentes roles que se han creado en el sistema, y a los cuales se vincularan los usuarios.	Tabla
Rolespermisorecursos	En esta tabla se establecen los Permisos que tendrá un Rol sobre los diferentes Recursos existentes en el	Tabla

Continuación Tabla 13.

NOMBRE	DETALLE	TIPO
	sistema.	
Sedes	Aquí se almacena el nombre de cada una de las sedes pertenecientes a la UDENAR	Tabla
Softwareaulas	Se almacena la información del software que está instalado en cada una de las aulas de informática, estos datos serán tenidos en cuenta para la elaboración de los horarios de las clases que se dictaran en cada una de ellas.	Tabla
Solicitudes	Esta tabla contiene la información de las solicitudes de los funcionarios de la institución.	Tabla
Ubicación	En esta tabla se especifica la sede en que se encuentra ubicada cada una de las dependencias existentes en la universidad.	Tabla
Usuarios	Esta tabla almacena la información básica de los usuarios que existen en el sistema de información integrado de la UIT.	Tabla

3.5.1 Detalles tabla aulas.

Figura 14. Detalles tabla aulas

Name	Data Type	Allow Nulls
 id	int	<input type="checkbox"/>
nom_aula	varchar(20)	<input type="checkbox"/>
responsable	int	<input checked="" type="checkbox"/>
admin_horario	bit	<input type="checkbox"/>
internet	bit	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Keys (1)
 PK_aulas (Primary Key, Clustered: id)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
 FK_aulas_usuarios (codigo)
Triggers (0)

SQL:
 CREATE TABLE [dbo].[aulas] (
 [id] INT NOT NULL,
 [nom_aula] VARCHAR (20) NOT NULL,

```

[responsable] INT NULL,
[admin_horario] BIT NOT NULL,
[internet] BIT NULL,
CONSTRAINT [PK_aulas] PRIMARY KEY CLUSTERED ([id] ASC),
CONSTRAINT [FK_aulas_usuarios] FOREIGN KEY ([responsable])
REFERENCES [dbo].[usuarios] ([código])
);

```

3.5.2 Detalles tabla dependencias

Figura 15. Detalles tabla dependencias

Name	Data Type	Allow Nulls	Defau
Id	int	<input type="checkbox"/>	
dependencia	varchar(20)	<input type="checkbox"/>	
		<input type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```

SQL:
CREATE TABLE [dbo].[dependencias] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [dependencia] VARCHAR (20) NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);

```

3.5.3 Detalles tabla dispositivos equipo.

Figura 16. Detalles tabla dispositivos equipo

Name	Data Type	Allow Nulls
Id	decimal(4,0)	<input type="checkbox"/>
nequipo	decimal(5,0)	<input type="checkbox"/>
dispositivo	varchar(20)	<input type="checkbox"/>
marca	varchar(20)	<input type="checkbox"/>
descripcion	varchar(MAX)	<input type="checkbox"/>

Keys (1)
PK_dispositivosequipo (Primary Key, Clustered: Id)

Check Constraints (0)

Indexes (0)

Foreign Keys (1)
FK_dispositivosequipo_equipos (n_equipo)

Triggers (0)

SQL:

```
CREATE TABLE [dbo].[dispositivosequipo] (  
    [Id]          DECIMAL (4) IDENTITY (1, 1) NOT NULL,  
    [nequipo]    DECIMAL (5) NOT NULL,  
    [dispositivo] VARCHAR (20) NOT NULL,  
    [marca]      VARCHAR (20) NOT NULL,  
    [descripcion] VARCHAR (MAX) NOT NULL,  
    CONSTRAINT [PK_dispositivosequipo] PRIMARY KEY CLUSTERED ([Id]  
ASC),  
    CONSTRAINT [FK_dispositivosequipo_equipos] FOREIGN KEY ([nequipo])  
REFERENCES [dbo].[equipos] ([n_equipo])  
);
```

3.5.4 Detalles tabla equipos.

Figura 17. Detalles tabla equipos

Name	Data Type	Allow Nulls	Default
n_equipo	decimal(5,0)	<input type="checkbox"/>	
Id_sede	int	<input type="checkbox"/>	
Id_dep	int	<input type="checkbox"/>	
ubicacion	varchar(50)	<input type="checkbox"/>	
nui	decimal(10,0)	<input type="checkbox"/>	((0))
tipo_equipo	varchar(30)	<input type="checkbox"/>	('N/A')
marca	varchar(20)	<input type="checkbox"/>	('N/A')
modelo	varchar(20)	<input type="checkbox"/>	('N/A')
serial	varchar(30)	<input type="checkbox"/>	('N/A')
garantia	date	<input type="checkbox"/>	('1/01/1900')
prestamo	bit	<input type="checkbox"/>	((0))
marcaf fuente	varchar(20)	<input type="checkbox"/>	('N/A')
marcaboard	varchar(20)	<input type="checkbox"/>	('N/A')
Modeloboard	varchar(20)	<input type="checkbox"/>	('N/A')
Marcap procesador	varchar(20)	<input type="checkbox"/>	('N/A')
Modelo procesador	varchar(20)	<input type="checkbox"/>	('N/A')
descripcion proc	varchar(30)	<input type="checkbox"/>	('N/A')
marcadisco	varchar(20)	<input type="checkbox"/>	('N/A')
capacidad disco	varchar(6)	<input type="checkbox"/>	('N/A')
integrada tarjered	bit	<input type="checkbox"/>	((0))
mac	varchar(17)	<input type="checkbox"/>	('N/A')
integrada tarjeredinalam	bit	<input type="checkbox"/>	((0))
macinalam	varchar(17)	<input type="checkbox"/>	('N/A')
integrada tarjevideo	bit	<input type="checkbox"/>	((0))
integrada tarsonido	bit	<input type="checkbox"/>	((0))
marcamemo1	varchar(20)	<input type="checkbox"/>	('N/A')
tipomemo1	varchar(5)	<input type="checkbox"/>	('N/A')
capacidad memo1	varchar(10)	<input type="checkbox"/>	('N/A')
marcamemo2	varchar(20)	<input type="checkbox"/>	('N/A')
tipomemo2	varchar(5)	<input type="checkbox"/>	('N/A')
capacidad memo2	varchar(10)	<input type="checkbox"/>	('N/A')
aula	int	<input type="checkbox"/>	((0))
observaciones	varchar(MAX)	<input type="checkbox"/>	('N/A')
estado	varchar(15)	<input type="checkbox"/>	('Funcionando')

- PK_equipos (Primary Key, Clustered: n_equipo)
 - CK_equipos_estado (estado)
 - Indexes (0)
 - FK_equipos_dependencias (Id)
 - FK_equipos_sedes (id_sede)
 - FK_equipos_aulas (id)
 - Triggers (0)

SQL:

```
CREATE TABLE [dbo].[equipos] (  
    [n_equipo]          DECIMAL (5) IDENTITY (1, 1) NOT NULL,  
    [ld_sede]          INT          NOT NULL,  
    [ld_dep]          INT          NOT NULL,  
    [ubicacion]       VARCHAR (50) NOT NULL,  
    [nui]             DECIMAL (10) CONSTRAINT  
[DF__equipos__nui__67A95F59] DEFAULT ((0)) NOT NULL,  
    [tipo_equipo]     VARCHAR (30) CONSTRAINT  
[DF__tmp_ms_xx_tipo__73DA2C14] DEFAULT ('N/A') NOT NULL,  
    [marca]          VARCHAR (20) CONSTRAINT  
[DF__tmp_ms_xx_marca__74CE504D] DEFAULT ('N/A') NOT NULL,  
    [modelo]         VARCHAR (20) CONSTRAINT  
[DF__tmp_ms_xx_model__75C27486] DEFAULT ('N/A') NOT NULL,  
    [serial]         VARCHAR (30) CONSTRAINT  
[DF__tmp_ms_xx_seria__76B698BF] DEFAULT ('N/A') NOT NULL,  
    [garantia]       DATE          CONSTRAINT  
[DF__tmp_ms_xx_garan__697C9932] DEFAULT ('1/01/1900') NOT NULL,  
    [prestamo]       BIT          CONSTRAINT [DF_equipos_prestamo]  
DEFAULT ((0)) NOT NULL,  
    [marcafuente]    VARCHAR (20) CONSTRAINT  
[DF__tmp_ms_xx_marca__77AABCF8] DEFAULT ('N/A') NOT NULL,  
    [marcaboard]     VARCHAR (20) CONSTRAINT  
[DF__tmp_ms_xx_marca__789EE131] DEFAULT ('N/A') NOT NULL,  
    [Modeloboard]    VARCHAR (20) CONSTRAINT  
[DF__tmp_ms_xx_Model__7993056A] DEFAULT ('N/A') NOT NULL,  
    [Marcaprocesador] VARCHAR (20) CONSTRAINT  
[DF__tmp_ms_xx_Marca__7A8729A3] DEFAULT ('N/A') NOT NULL,  
    [Modeloprocesador] VARCHAR (20) CONSTRAINT  
[DF__tmp_ms_xx_Model__7B7B4DDC] DEFAULT ('N/A') NOT NULL,  
    [descripcionproc] VARCHAR (30) CONSTRAINT  
[DF__tmp_ms_xx_descr__7C6F7215] DEFAULT ('N/A') NOT NULL,  
    [marcadiscod]    VARCHAR (20) CONSTRAINT  
[DF__tmp_ms_xx_marca__7D63964E] DEFAULT ('N/A') NOT NULL,  
    [capacidaddisco] VARCHAR (6)  CONSTRAINT  
[DF__tmp_ms_xx_capac__7E57BA87] DEFAULT ('N/A') NOT NULL,  
    [integratarjered] BIT          CONSTRAINT  
[DF__tmp_ms_xx_integ__7F4BDEC0] DEFAULT ((0)) NOT NULL,  
    [mac]            VARCHAR (17) CONSTRAINT  
[DF__tmp_ms_xx_e_mac__004002F9] DEFAULT ('N/A') NOT NULL,  
    [integratarjeredinalam] BIT    CONSTRAINT  
[DF__tmp_ms_xx_integ__01342732] DEFAULT ((0)) NOT NULL,  
    [macinalam]      VARCHAR (17) CONSTRAINT  
[DF__tmp_ms_xx_macin__02284B6B] DEFAULT ('N/A') NOT NULL,  
    [integratarjevideo] BIT        CONSTRAINT
```

```

[DF__tmp_ms_xx_integ__031C6FA4] DEFAULT ((0)) NOT NULL,
[integradatarsonido] BIT CONSTRAINT
[DF__tmp_ms_xx_integ__041093DD] DEFAULT ((0)) NOT NULL,
[marcamemo1] VARCHAR (20) CONSTRAINT
[DF__tmp_ms_xx_marca__0504B816] DEFAULT ('N/A') NOT NULL,
[tipomemo1] VARCHAR (5) CONSTRAINT
[DF__tmp_ms_xx_tipom__05F8DC4F] DEFAULT ('N/A') NOT NULL,
[capacidadmemo1] VARCHAR (10) CONSTRAINT
[DF__tmp_ms_xx_capac__06ED0088] DEFAULT ('N/A') NOT NULL,
[marcamemo2] VARCHAR (20) CONSTRAINT
[DF__tmp_ms_xx_marca__07E124C1] DEFAULT ('N/A') NOT NULL,
[tipomemo2] VARCHAR (5) CONSTRAINT
[DF__tmp_ms_xx_tipom__08D548FA] DEFAULT ('N/A') NOT NULL,
[capacidadmemo2] VARCHAR (10) CONSTRAINT
[DF__tmp_ms_xx_capac__09C96D33] DEFAULT ('N/A') NOT NULL,
[aula] INT CONSTRAINT [DF__equipos__aula__74B941B4]
DEFAULT ((0)) NOT NULL,
[observaciones] VARCHAR (MAX) CONSTRAINT
[DF__tmp_ms_xx_obser__0ABD916C] DEFAULT ('N/A') NOT NULL,
[estado] VARCHAR (15) CONSTRAINT
[DF__tmp_ms_xx_estad__0BB1B5A5] DEFAULT ('Funcionando') NOT NULL,
CONSTRAINT [PK_equipos] PRIMARY KEY CLUSTERED ([n_equipo] ASC),
CONSTRAINT [FK_equipos_dependencias] FOREIGN KEY ([Id_dep])
REFERENCES [dbo].[dependencias] ([Id]),
CONSTRAINT [FK_equipos_sedes] FOREIGN KEY ([Id_sede]) REFERENCES
[dbo].[sedes] ([id_sede]),
CONSTRAINT [FK_equipos_aulas] FOREIGN KEY ([aula]) REFERENCES
[dbo].[aulas] ([id]),
CONSTRAINT [CK_equipos_estado] CHECK ([estado] = 'Funcionando'
OR [estado] = 'Mantenimiento'
OR [estado] = 'Dado de baja')
);

```

3.5.6 Detalles tabla horarios aulas.

Figura 18. Detalles tabla horarios aulas

Name	Data Type	Allow Nulls	Default
id	int	<input type="checkbox"/>	
nomcurso	varchar(20)	<input type="checkbox"/>	
programa	int	<input type="checkbox"/>	
docentecurso	varchar(30)	<input type="checkbox"/>	
aula	int	<input type="checkbox"/>	
n_horas	int	<input type="checkbox"/>	
dia	varchar(10)	<input type="checkbox"/>	
hora	time(1)	<input type="checkbox"/>	

▲ **Keys** (1)
PK_horariosalulas (Primary Key, Clustered: id)

Check Constraints (0)

Indexes (0)

▲ **Foreign Keys** (2)
FK_horariosalulas_aulas (id)
FK_horariosalulas_programas (id)

Triggers (0)

SQL:

```
CREATE TABLE [dbo].[horariosalulas] (  
    [id] INT IDENTITY (1, 1) NOT NULL,  
    [nomcurso] VARCHAR (20) NOT NULL,  
    [programa] INT NOT NULL,  
    [docentecurso] VARCHAR (30) NOT NULL,  
    [aula] INT NOT NULL,  
    [n_horas] INT NOT NULL,  
    [dia] VARCHAR (10) NOT NULL,  
    [hora] TIME (1) NOT NULL,  
    CONSTRAINT [PK_horariosalulas] PRIMARY KEY CLUSTERED ([id] ASC),  
    CONSTRAINT [FK_horariosalulas_aulas] FOREIGN KEY ([aula])  
REFERENCES [dbo].[aulas] ([id]),  
    CONSTRAINT [FK_horariosalulas_programas] FOREIGN KEY ([programa])  
REFERENCES [dbo].[programas] ([id])  
);
```

3.5.7 Detalles tabla horariosm.

Figura 19. Detalles tabla horariosm

Name	Data Type	Allow Nulls	Default
codigo	int	<input type="checkbox"/>	
diahora	varchar(20)	<input type="checkbox"/>	
horario	varchar(20)	<input type="checkbox"/>	
		<input type="checkbox"/>	

▲ Keys (1)
 PK_horariosm (Primary Key, Clustered: codigo, diahora)

Check Constraints (0)

Indexes (0)

▲ Foreign Keys (1)
 FK_horariosm_usuarios (codigo)

Triggers (0)

```
SQL:
CREATE TABLE [dbo].[horariosm] (
    [codigo] INT NOT NULL,
    [diahora] VARCHAR (20) NOT NULL,
    [horario] VARCHAR (20) NOT NULL,
    CONSTRAINT [PK_horariosm] PRIMARY KEY CLUSTERED ([codigo] ASC,
[diahora] ASC),
    CONSTRAINT [FK_horariosm_usuarios] FOREIGN KEY ([codigo])
REFERENCES [dbo].[usuarios] ([codigo])
);
```


3.5.8 Detalles tabla mantenimientos equipos.

Figura 20. Detalles tabla mantenimientos equipos

Name	Data Type	Allow Nulls	Default
solicitud	int	<input checked="" type="checkbox"/>	
descripcion	varchar(MAX)	<input type="checkbox"/>	
tipo	varchar(20)	<input type="checkbox"/>	
clasificacion	varchar(20)	<input type="checkbox"/>	
fecha	smalldatetime	<input type="checkbox"/>	
responsable	int	<input type="checkbox"/>	
nequipo	decimal(5,0)	<input type="checkbox"/>	

- ▲ **Keys** (1)
 - PK_mantenimientosequipos (Primary Key, Clustered: id)
- ▲ **Check Constraints** (1)
 - CK_mantenimientosequipos_tipo (tipo)
- Indexes** (0)
- ▲ **Foreign Keys** (3)
 - FK_mantenimientosequipos_solicitudes (id)
 - FK_mantenimientosequipos_usuarios (codigo)
 - FK_mantenimientosequipos_equipos (n_equipo)
- Triggers** (0)

SQL:

```
CREATE TABLE [dbo].[mantenimientosequipos] (  
    [id]          DECIMAL (5) IDENTITY (1, 1) NOT NULL,  
    [solicitud]   INT          NULL,  
    [descripcion] VARCHAR (MAX) NOT NULL,  
    [tipo]        VARCHAR (20) NOT NULL,  
    [clasificacion] VARCHAR (20) NOT NULL,  
    [fecha]       SMALLDATETIME NOT NULL,  
    [responsable] INT          NOT NULL,  
    [nequipo]     DECIMAL (5) NOT NULL,  
    CONSTRAINT [PK_mantenimientosequipos] PRIMARY KEY CLUSTERED ([id]  
ASC),  
    CONSTRAINT [FK_mantenimientosequipos_solicitudes] FOREIGN KEY  
([solicitud]) REFERENCES [dbo].[solicitudes] ([id]),  
    CONSTRAINT [FK_mantenimientosequipos_usuarios] FOREIGN KEY  
([responsable]) REFERENCES [dbo].[usuarios] ([codigo]),  
    CONSTRAINT [FK_mantenimientosequipos_equipos] FOREIGN KEY ([nequipo])  
REFERENCES [dbo].[equipos] ([n_equipo]),  
    CONSTRAINT [CK_mantenimientosequipos_tipo] CHECK ([tipo]='correctivo' OR  
[tipo]='preventivo')  
);
```

3.5.9 Detalles tabla permisos.

Figura 21. Detalles tabla permisos

	Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	permiso	varchar(20)	<input type="checkbox"/>


▲ **Keys (1)**
 <unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

SQL:

```
CREATE TABLE [dbo].[permisos] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [permiso] VARCHAR (20) NOT NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

3.5.10 Detalles tabla préstamos.

Figura 22. Detalles tabla préstamos

	Name	Data Type	Allow Nulls	Default
	id	decimal(10,0)	<input type="checkbox"/>	
	documento	varchar(10)	<input type="checkbox"/>	
	nombre	varchar(30)	<input type="checkbox"/>	
	equipo	decimal(5,0)	<input type="checkbox"/>	
	fechahora	datetime	<input type="checkbox"/>	

▲ **Keys** (1)
 PK_prestamos (Primary Key, Clustered: id)

Check Constraints (0)

Indexes (0)

▲ **Foreign Keys** (1)
 FK_prestamos_equipos (n_equipo)

Triggers (0)

SQL:

```
CREATE TABLE [dbo].[prestamos] (  
    [id]      DECIMAL (10) IDENTITY (1, 1) NOT NULL,  
    [documento] VARCHAR (10) NOT NULL,  
    [nombre]  VARCHAR (30) NOT NULL,  
    [equipo]  DECIMAL (5) NOT NULL,  
    [fechahora] DATETIME  NOT NULL,  
    CONSTRAINT [PK_prestamos] PRIMARY KEY CLUSTERED ([id] ASC),  
    CONSTRAINT [FK_prestamos_equipos] FOREIGN KEY ([equipo]) REFERENCES  
    [dbo].[equipos] ([n_equipo])  
);
```

3.5.11 Detalles tabla préstamosotros.

Figura 23 Detalles tabla préstamosotros

Name	Data Type	Allow Nulls	Default
id	decimal(10,0)	<input type="checkbox"/>	
correo	varchar(50)	<input type="checkbox"/>	
nombre	varchar(50)	<input type="checkbox"/>	
equipo	decimal(5,0)	<input type="checkbox"/>	
fechahora	datetime	<input type="checkbox"/>	
estado	varchar(15)	<input type="checkbox"/>	('Reservado')
observacionesdev	varchar(MAX)	<input type="checkbox"/>	('N/A')
calificacion	varchar(15)	<input type="checkbox"/>	('N/A')
observacionescalif	varchar(MAX)	<input type="checkbox"/>	('N/A')
entrega	int	<input checked="" type="checkbox"/>	
recibe	int	<input checked="" type="checkbox"/>	

Keys (1)
 PKprestamosotros (Primary Key, Clustered: id)

Check Constraints (0)

Indexes (0)

Foreign Keys (3)
 FKprestamosotros Equipos (n_equipos)
 FKprestamosotros usuarios (codigo)
 FKprestamosotros usuarios1 (codigo)

Triggers (0)

SQL:

```

CREATE TABLE [dbo].[prestamosotros] (
    [id]          DECIMAL (10) IDENTITY (1, 1) NOT NULL,
    [correo]      VARCHAR (50) NOT NULL,
    [nombre]     VARCHAR (50) NOT NULL,
    [equipo]     DECIMAL (5) NOT NULL,
    [fechahora]  DATETIME NOT NULL,
    [estado]     VARCHAR (15) CONSTRAINT
    [DF__prestamos__estad__542254F0] DEFAULT ('Reservado') NOT NULL,
    [observacionesdev] VARCHAR (MAX) CONSTRAINT
    [DF__prestamos__obser__6458BCB9] DEFAULT ('N/A') NOT NULL,
    [calificacion] VARCHAR (15) CONSTRAINT
  
```

```

[DF__prestamos__calif__654CE0F2] DEFAULT ('N/A') NOT NULL,
 [observacionescalif] VARCHAR (MAX) CONSTRAINT
[DF__prestamos__obser__6641052B] DEFAULT ('N/A') NOT NULL,
 [entrega] INT NULL,
 [recibe] INT NULL,
 CONSTRAINT [PK_prestamosotros] PRIMARY KEY CLUSTERED ([id] ASC),
 CONSTRAINT [FK_prestamosotros_equipos] FOREIGN KEY ([equipo])
REFERENCES [dbo].[equipos] ([n_equipo]),
 CONSTRAINT [FK_prestamosotros_usuarios] FOREIGN KEY ([entrega])
REFERENCES [dbo].[usuarios] ([codigo]),
 CONSTRAINT [FK_prestamosotros_usuarios1] FOREIGN KEY ([recibe])
REFERENCES [dbo].[usuarios] ([codigo])
);

```

3.5.12 Detalles tabla programas.

Figura 24. Detalles tabla programas

Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
nombre	varchar(30)	<input type="checkbox"/>

Keys (2)
 PK_programas (Primary Key, Clustered: id)
 AK_programas (nombre)

Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```

SQL:
CREATE TABLE [dbo].[programas] (
 [id] INT IDENTITY (1, 1) NOT NULL,
 [nombre] VARCHAR (30) NOT NULL,
 CONSTRAINT [PK_programas] PRIMARY KEY CLUSTERED ([id] ASC),
 CONSTRAINT [AK_programas] UNIQUE NONCLUSTERED ([nombre] ASC)
);

```

3.5.12 Detalles tabla recursos.

Figura 25. Detalles tabla recursos

Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
recurso	varchar(30)	<input type="checkbox"/>

Keys (1)
PK_recursos (Primary Key, Clustered: id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

SQL:

```
CREATE TABLE [dbo].[recursos] (  
    [id] INT IDENTITY (1, 1) NOT NULL,  
    [recurso] VARCHAR (30) NOT NULL,  
    CONSTRAINT [PK_recursos] PRIMARY KEY CLUSTERED ([id] ASC)  
);
```

3.5.13 Detalles tabla roles.

Figura 26. Detalles tabla roles

Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
rol	varchar(20)	<input type="checkbox"/>

Keys (1)
<unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
SQL:
CREATE TABLE [dbo].[roles] (
  [Id] INT IDENTITY (1, 1) NOT NULL,
  [rol] VARCHAR (20) NOT NULL,
  PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

3.5.14 Detalles tabla rolespermisorecursos

Figura 27. Detalles tabla rolespermisorecursos

Name	Data Type	Allow Nulls	Default
rol	int	<input type="checkbox"/>	
permiso	int	<input type="checkbox"/>	
recursos	int	<input type="checkbox"/>	

- ▲ **Keys (1)**
 - <unnamed> (Primary Key, Clustered: recursos, permiso, rol)
- Check Constraints (0)**
- Indexes (0)**
- ▲ **Foreign Keys (3)**
 - FK_rolespermiso_roles (Id)
 - FK_rolespermiso_permisos (Id)
 - FK_rolespermisorecursos_recursos (id)
- Triggers (0)**

```
SQL:
CREATE TABLE [dbo].[rolespermisorecursos] (
  [rol] INT NOT NULL,
  [permiso] INT NOT NULL,
  [recursos] INT NOT NULL,
  PRIMARY KEY CLUSTERED ([recursos] ASC, [permiso] ASC, [rol] ASC),
  CONSTRAINT [FK_rolespermiso_roles] FOREIGN KEY ([rol]) REFERENCES
[dbo].[roles] ([Id]),
  CONSTRAINT [FK_rolespermiso_permisos] FOREIGN KEY ([permiso])
REFERENCES [dbo].[permisos] ([Id]),
  CONSTRAINT [FK_rolespermisorecursos_recursos] FOREIGN KEY ([recursos])
REFERENCES [dbo].[recursos] ([id])
);
```

3.5.15 Detalles tabla sedes.

Figura 28. Detalles tabla sedes

Name	Data Type	Allow Nulls
id_sede	int	<input type="checkbox"/>
nom_sede	varchar(20)	<input type="checkbox"/>

Keys (1)
PK_sedes (Primary Key, Clustered: id_sede)

Check Constraints (0)

Indexes (0)

Foreign Keys (0)

Triggers (0)

SQL:

```
CREATE TABLE [dbo].[sedes] (  
    [id_sede] INT IDENTITY (1, 1) NOT NULL,  
    [nom_sede] VARCHAR (20) NOT NULL,  
    CONSTRAINT [PK_sedes] PRIMARY KEY CLUSTERED ([id_sede] ASC)  
);
```

3.5.16 Detalles tabla softwareaulas.

Figura 29. Detalles tabla softwareaulas

Name	Data Type	Allow Nulls	Default
id	int	<input type="checkbox"/>	
aula	int	<input type="checkbox"/>	
nombre_software	varchar(30)	<input type="checkbox"/>	

Keys (1)
PK_softwareaulas (Primary Key, Clustered: id)

Check Constraints (0)

Indexes (0)

Foreign Keys (1)
FK_softwareaulas_aulas (id)

Triggers (0)


```

SQL:
CREATE TABLE [dbo].[softwareaulas] (
  [id]          INT          IDENTITY (1, 1) NOT NULL,
  [aula]       INT          NOT NULL,
  [nombre_software] VARCHAR (30) NOT NULL,
  CONSTRAINT [PK_softwareaulas] PRIMARY KEY CLUSTERED ([id] ASC),
  CONSTRAINT [FK_softwareaulas_aulas] FOREIGN KEY ([aula])
REFERENCES [dbo].[aulas] ([id])
);

```

3.5.17 Detalles tabla solicitudes.

Figura 30. Detalles tabla solicitudes

Name	Data Type	Allow Nulls	Default
id	int	<input type="checkbox"/>	
id_sede	int	<input type="checkbox"/>	
id_dep	int	<input type="checkbox"/>	
ubicacion	varchar(30)	<input type="checkbox"/>	
descripcion	varchar(200)	<input type="checkbox"/>	
nombre_solic	varchar(30)	<input type="checkbox"/>	
solocitante	varchar(30)	<input type="checkbox"/>	
fechahora	smalldatetime	<input type="checkbox"/>	
responsable	int	<input checked="" type="checkbox"/>	
calificacion	varchar(10)	<input type="checkbox"/>	('N/A')
observaciones	varchar(MAX)	<input type="checkbox"/>	('N/A')
estado	varchar(10)	<input type="checkbox"/>	('Pendiente')
medio_solic	varchar(10)	<input type="checkbox"/>	('Web')

Keys (1)
 PK_solicitudes (Primary Key, Clustered: id)

Check Constraints (2)
 CK_solicitudes_estado (estado)
 CK_solicitudes_Medio (medio_solic)

Indexes (0)

Foreign Keys (3)
 FK_solicitudes_dependencias (Id)
 FK_solicitudes_sedes (id_sede)
 FK_solicitudes_usuarios (codigo)

SQL:

```
CREATE TABLE [dbo].[solicitudes] (  
    [id]          INT          IDENTITY (1, 1) NOT NULL,  
    [id_sede]    INT          NOT NULL,  
    [id_dep]     INT          NOT NULL,  
    [ubicacion]  VARCHAR (30) NOT NULL,  
    [descripcion] VARCHAR (200) NOT NULL,  
    [nombre_solic] VARCHAR (30) NOT NULL,  
    [solocitante] VARCHAR (30) NOT NULL,  
    [fechahora]  SMALLDATETIME NOT NULL,  
    [responsable] INT          NULL,  
    [calificacion] VARCHAR (10) CONSTRAINT [DF__solicitud__calif__318258D2]  
    DEFAULT ('N/A') NOT NULL,  
    [observaciones] VARCHAR (MAX) CONSTRAINT  
    [DF__solicitud__obser__32767D0B] DEFAULT ('N/A') NOT NULL,  
    [estado]     VARCHAR (10) CONSTRAINT  
    [DF__solicitud__estad__22401542] DEFAULT ('Pendiente') NOT NULL,  
    [medio_solic] VARCHAR (10) CONSTRAINT [DF__solicitud__default_medio]  
    DEFAULT ('Web') NOT NULL,  
    CONSTRAINT [PK_solicitudes] PRIMARY KEY CLUSTERED ([id] ASC),  
    CONSTRAINT [FK_solicitudes_dependencias] FOREIGN KEY ([id_dep])  
    REFERENCES [dbo].[dependencias] ([Id]),  
    CONSTRAINT [FK_solicitudes_sedes] FOREIGN KEY ([id_sede])  
    REFERENCES [dbo].[sedes] ([id_sede]),  
    CONSTRAINT [FK_solicitudes_usuarios] FOREIGN KEY ([responsable])  
    REFERENCES [dbo].[usuarios] ([codigo]),  
    CONSTRAINT [CK_solicitudes_estado] CHECK ([estado]='Terminado' OR  
    [estado]='Pendiente'),  
    CONSTRAINT [CK_solicitudes_Medio] CHECK ([medio_solic]='Web' OR  
    [medio_solic]='Correo' OR [medio_solic]='Telefono' OR [medio_solic]='Link' OR  
    [medio_solic]='Presencial')  
);
```

3.5.18 Detalles tabla ubicación.

Figura 31. Detalles tabla ubicación

Name	Data Type	Allow Nulls	Default
Id_sede	int	<input type="checkbox"/>	
Id_dep	int	<input type="checkbox"/>	

▲ **Keys (1)**
PK_ubicacion (Primary Key, Clustered: Id_sede, Id_dep)

Check Constraints (0)

Indexes (0)

▲ **Foreign Keys (2)**
FK_ubicacion_dependencias (Id)
FK_ubicacion_sedes (id_sede)

Triggers (0)

SQL:

```
CREATE TABLE [dbo].[ubicacion] (  
    [Id_sede] INT NOT NULL,  
    [Id_dep] INT NOT NULL,  
    CONSTRAINT [PK_ubicacion] PRIMARY KEY CLUSTERED ([Id_sede] ASC,  
[Id_dep] ASC),  
    CONSTRAINT [FK_ubicacion_dependencias] FOREIGN KEY ([Id_dep])  
REFERENCES [dbo].[dependencias] ([Id]),  
    CONSTRAINT [FK_ubicacion_sedes] FOREIGN KEY ([Id_sede])  
REFERENCES [dbo].[sedes] ([id_sede])  
);
```

3.5.19 Detalles tabla usuarios.

Figura 32. Detalles tabla usuarios

Name	Data Type	Allow Nulls	Default
codigo	int	<input type="checkbox"/>	
nombre_apellido	varchar(50)	<input type="checkbox"/>	
correo	varchar(50)	<input type="checkbox"/>	
telefono	nvarchar(10)	<input type="checkbox"/>	
programa	nvarchar(30)	<input type="checkbox"/>	
vigencia	nvarchar(10)	<input type="checkbox"/>	('Vigente')
Rol	int	<input type="checkbox"/>	
tipovinculacion	varchar(20)	<input type="checkbox"/>	
		<input type="checkbox"/>	

Keys (1)
PK_usuarios (Primary Key, Clustered: codigo)

Check Constraints (2)
CK_usuarios_tipovinculacion (tipovinculacion)
CK_usuarios_vigencia (vigencia)

Indexes (0)

Foreign Keys (0)

Triggers (0)

SQL:

```
CREATE TABLE [dbo].[usuarios] (  
    [codigo] INT NOT NULL,  
    [nombre_apellido] VARCHAR (50) NOT NULL,  
    [correo] VARCHAR (50) NOT NULL,  
    [telefono] NVARCHAR (10) NOT NULL,  
    [programa] NVARCHAR (30) NOT NULL,  
    [vigencia] NVARCHAR (10) CONSTRAINT  
    [DF__usuarios__vigenc__2CBDA3B5] DEFAULT ('Vigente') NOT NULL,  
    [Rol] INT NOT NULL,  
    [tipovinculacion] VARCHAR (20) NOT NULL,  
    CONSTRAINT [PK_usuarios] PRIMARY KEY CLUSTERED ([codigo] ASC),  
    CONSTRAINT [CK_usuarios_tipovinculacion] CHECK ([tipovinculacion]='Tecnico'  
    OR [tipovinculacion]='MedioT' OR [tipovinculacion]='Funcionario'),  
    CONSTRAINT [CK_usuarios_vigencia] CHECK ([vigencia]='No_vigente' OR  
    [vigencia]='Vigente')  
);
```

3.6 TARJETAS CRC

La metodología eXtreme Programing no describe los sistemas mediante diagramas UML, en cambio, los reemplaza por otros recursos como tarjetas CRC (Contenido, Responsabilidad y Colaboración), siempre y cuando sean beneficiosas para el sistema y no sean difíciles de mantener en el tiempo.

Las ventajas que ofrecen este tipo de tarjetas, es que nos permiten desarrollar solamente la funcionalidades requeridas y no crear código inservible. Por ende, se recomienda la utilización de programación orientada a objetos con sus respectivos métodos debido a que generalmente cada tarjeta CRC representa un objeto a desarrollar.

3.6.1 Tarjeta CRC menú principal.

Tabla 14. Tarjeta CRC menú principal

MENÚ PRINCIPAL	
Funcionalidades	Colaboradores
<p>Mostrar las opciones para ingresar al sistema tanto para los funcionarios y docentes de la universidad, como para los trabajadores de la UIT.</p> <p>Mostrar las opciones para realizar reserva de equipos que la UIT ha puesto a disposición del público de la institución</p> <p>Validación de los datos de autenticación que los usuarios ingresen.</p>	<p>Funcionario y docentes UDENAR.</p> <p>Monitores.</p> <p>Administradores del sistema</p>

3.6.2 Tarjeta CRC módulo usuarios.

Tabla 15. Tarjeta CRC módulo usuarios

MÓDULO USUARIOS	
Funcionalidades	Colaboradores
<p>Crear, Editar, Deshabilitar los usuarios que estarán en contactos con el sistema de información integrado.</p> <p>Crear, Editar, Eliminar roles a los cuales se asignaran los monitores y funcionarios del sistema.</p>	<p>Usuario Final (Monitores cuyo rol tenga permisos para acceder a estas opciones)</p> <p>Administrador del Sistema.</p>

3.6.3 Tarjeta CRC módulo solicitudes

Tabla 16. Tarjeta CRC módulo solicitudes

MÓDULO SOLICITUDES	
Funcionalidades	Colaboradores
Realizar solicitudes por parte de los funcionarios y docentes de la institución. Permitir que los monitores o los administrativos atiendan las solicitudes realizadas. Calificar las solicitudes que hayan sido finalizadas. Eliminar las solicitudes que no hayan sido desarrolladas y que ya no sean necesarias. Consultar el historial de las últimas solicitudes de asistencia realizadas	Funcionarios y docentes de la universidad. Usuario Final (Monitores cuyo rol tenga permisos para acceder a estas opciones) Administrador del Sistema

3.6.4 Tarjeta CRC módulo inventario.

Tabla 17. Tarjeta CRC módulo inventario

MÓDULO INVENTARIO	
Funcionalidades	Colaboradores
Crear (individual o por lotes), Editar, Dar de Baja a los equipos de cómputo y a fines que están a cargo de la UIT. Poner en mantenimiento los equipos cuya reparación sea un poco dispendiosa. Registrar mantenimientos realizados al equipo.	Usuario Final (Monitores cuyo rol tenga permisos para acceder a estas opciones) Administrador del Sistema

3.6.5 Tarjeta CRC módulo horarios monitores.

Tabla 18. Tarjeta CRC módulo horarios monitores

MÓDULO HORARIOS MONITORES	
Funcionalidades	Colaboradores
Crear, Editar, Eliminar Horarios de acuerdo a la necesidad que sea determinada por la UIT para administrar las jornadas laborales de cada uno de los monitores.	Usuario Final (Monitores cuyo rol tenga permisos para acceder a estas opciones) Administrador del Sistema

3.6.6 Tarjeta CRC módulo aulas

Tabla 19. Tarjeta CRC módulo aulas

MÓDULO AULAS	
Funcionalidades	Colaboradores
Crear, Editar, Eliminar Aulas, registrar o quitar software instalado en cada una de las aulas, Crear, Editar, Eliminar horarios de las aulas de informática sin que haya cruces en cuanto a aulas y/o a docentes.	Usuario Final (Secretaria, Monitores cuyo rol tenga permisos para acceder a estas opciones) Administrador del Sistema

3.6.7 Tarjeta CRC módulo préstamos.

Tabla 20. Tarjeta CRC módulo préstamos

MÓDULO PRESTAMOS	
Funcionalidades	Colaboradores
Reservar los equipos que estén disponibles para el préstamo al personal perteneciente a la UDENAR en los horarios establecidos para el acceso a internet. Reservar, tomar prestado y hacer la devolución de los equipos que están habilitados para préstamo, como video beams, micrófonos, parlantes etc. Los cuales se encuentren disponibles, se podrá llevar el control y seguimiento del préstamo, hasta su devolución.	Usuario Final (Monitores cuyo rol tenga permisos para acceder a estas opciones) Administrador del Sistema

3.6.8 Tarjeta CRC módulo reportes.

Tabla 21. Tarjeta CRC módulo reportes

MÓDULO REPORTES	
Funcionalidades	Colaboradores
Crear reportes de asistencias solicitadas a la UIT. Crear reportes de Asistencias realizadas a la UIT agrupado por dependencias y clasificándolo por calificación y por el funcionario que atendió dicho proceso. Crear reportes de Mantenimientos realizados a los equipos que están a cargo de la UIT, deben tener la opción de filtrar los datos de acuerdo a los criterios que los funcionarios de la dependencia consideren.	Usuario Final (Monitores cuyo rol tenga permisos para acceder a estas opciones) Administrador del Sistema

3.7 FASE DE PRUEBAS

Si hay algo importante en la metodología eXtreme Programming, es las pruebas que se realizan a las aplicaciones generadas, esto con el fin de que los requerimientos presentado en las historias de usuario y las funcionalidades descritas en las tablas CRC se hayan cumplido a cabalidad.

Las pruebas se realizan sin tener en cuenta el código implementado, simplemente se basan en la expectativa de los usuarios finales, así como también se prueban los puntos de quiebre que puede tener el software.

Se hacen pruebas de aceptación, las cuales evalúan si se han satisfecho todos los requerimientos realizados por parte de los usuarios finales.

3.7.1 Prueba de aceptación: Hu1- acceso al sistema.

Tabla 22. Prueba de aceptación: Hu1- acceso al sistema

Prueba de aceptación – HU1 - ACCESO AL SISTEMA	
Numero: 1	Historia de Usuario No. 1
<p>Descripción: La prueba se diseña con el fin de probar si la aplicación presenta las opciones adecuadas, para la autenticación de cada tipo de usuario que estar en contacto con el sistema, y también si la validación de los datos ingresados se realiza correctamente.</p>	
<p>Condiciones de Ejecución: El usuario deberá acceder aplicación web desde su navegador, se deberá mostrar un menú donde se muestre las opciones a las que él puede acceder, ya sea ingresar al sistema tanto para los usuarios del sistema, como para docentes y funcionarios o también para realizar reservas de equipos. Una vez seleccionada una de las opciones, el sistema queda a la espera de los datos de autenticación para realizar la respectiva validación.</p>	
<p>Entradas / Pasos de Ejecución: Entradas</p> <ol style="list-style-type: none"> 1. Selección Acceso a Asistencias 2. Selección de Acceso al Sistema UIT 3. Selección de reservas de equipos. 4. Usuario(correo/código) y clave correctos 5. Usuario (correo/código) y clave incorrectos, usuario (correo/código) incorrecto y clave correcta, usuario (correo/código) correcto y clave incorrecta, usuario (correo/código) y clave vacíos, usuario (correo/código) lleno y clave vacía, usuario (correo/código) vacío y clave llena. 	
<p>Resultados Esperados:</p> <ol style="list-style-type: none"> 1. Acceso a la página de login de la opción de asistencias. 2. Acceso a la página de login de la opción de Acceso al sistema UIT 3. Acceso a la página principal de reserva de equipos. 4. Re direccionamiento a las página principal del sistema o a la página de reservas de equipos o a la página principal de solicitudes según sea el caso. 5. Mensaje de advertencia indicando que el nombre de usuario o contraseña son incorrectos. 	<p>Resultados Obtenidos:</p> <ol style="list-style-type: none"> 1. Acceso a la página de login de la opción de asistencias. 2. Acceso a la página de login de la opción de Acceso al sistema UIT 3. Acceso a la página principal de reserva de equipos. 4. Re direccionamiento a las página principal del sistema o a la página de reservas de equipos o a la página principal de solicitudes según sea el caso. 5. Mensaje de advertencia indicando que el nombre de usuario o contraseña son incorrectos.
<p>Calificación de la Prueba: APROBADA ✓</p>	

3.7.2 Prueba de aceptación: Hu2 - usuarios y roles.

Tabla 23. Prueba de aceptación: Hu2 - usuarios y roles

Prueba de aceptación – HU2 - USUARIOS Y ROLES	
Numero: 2	Historia de Usuario No.2
<p>Descripción: Esta prueba está diseñada para garantizar la funcionalidad del módulo de usuarios y roles del sistema de información integrado de la UIT, exigiendo los campos que sean obligatorios y alertas cuando se lleven a cabo acciones.</p>	
<p>Condiciones de Ejecución: Un ves el usuario del sistema UIT ha superado la autenticación deberá acceder al menú Usuario, donde encontrara las opciones para crear, editar e inhabilitar a los usuarios del sistema, también tendrá disponible la opción de administrar los diferentes roles que se manejan en el sistema, de igual manera en esta última opción aparecerá una sección donde podrá editar las dependencias y las sedes de la institución.</p>	
<p>Entradas / Pasos de Ejecución: Entradas</p> <ol style="list-style-type: none"> 1. Seleccione Crear Usuario 2. Seleccione la opción Editar usuario 3. Seleccione la opción Roles, Permisos y Recursos 4. Controles diligenciados correctamente 5. Cajas de texto vacías 6. No haya datos en las listas desplegables. 	
<p>Resultados Esperados:</p> <ol style="list-style-type: none"> 1. Re-direccionar a la página donde se pueda diligenciar los datos del usuario y luego guardarlos en la base de datos. 2. Re-direccionar a la página donde se carguen los datos del usuario a editar, para diligenciar la información que se desea modificar y luego guardarlos en la base de datos. 3. Re-direccionar a la página donde se administran los roles, permisos, recursos, y también las dependencias de la Institución. 4. Guardar (crear, editar, Eliminar) en la base de datos. 5. Mensaje de Error informando que los datos están incompletos. 6. Controles que ejecutan acciones deshabilitados 	<p>Resultados Obtenidos:</p> <ol style="list-style-type: none"> 1. Re-direccionar a la página donde se pueda diligenciar los datos del usuario y luego guardarlos en la base de datos. 2. Re-direccionar a la página donde se carguen los datos del usuario a editar, para diligenciar la información que se desea modificar y luego guardarlos en la base de datos. 3. Re-direccionar a la página donde se administran los roles, permisos, recursos, y también las dependencias de la Institución. 4. Guardar (crear, editar, Eliminar) en la base de datos. 5. Mensaje de Error informando que los datos están incompletos. 6. Controles que ejecutan acciones deshabilitados
<p>Calificación de la Prueba: APROBADA ✓</p>	

3.7.3 Prueba de aceptación: HU3 - crear solicitudes.

Tabla 24. Prueba De aceptación: HU3 - crear solicitudes

Prueba de aceptación – HU3 - CREAR SOLICITUDES	
Numero: 3	Historia de Usuario No. 3
Descripción: Mediante esta prueba se verificara el funcionamiento correcto de la sección de creación de solicitudes, con el fin de garantizar su nivel de funcionalidad.	
Condiciones de Ejecución: Un ves el usuario del sistema UIT ha superado la autenticación deberá acceder al a la página donde podrá realizar las solicitudes, o también calificar las mismas que ya hayan sido terminadas.	
Entradas / Pasos de Ejecución: Entradas	
<ol style="list-style-type: none"> 1. No hay solicitudes por calificar 2. Hay solicitudes por calificar 3. Selecciona la opción “Eliminar Asistencias” 4. Selecciona la opción “Ver Historial de Solicitudes” 5. Datos correctamente diligenciados 6. Cajas de Texto Vacías 7. Calificación Insuficiente o Deficiente y no tiene Observación 	
Resultados Esperados:	Resultados Obtenidos:
<ol style="list-style-type: none"> 1. Mostrar las opciones para crear una nueva solicitud. 2. Mostrar las opciones para calificar una solicitud que ya está finalizada. 3. Mostrar la lista de solicitudes que puede eliminar, de las cuales el usuario podrá seleccionar una y esta será eliminada. 4. Mostrar un listado de las ultimas solicitudes realizadas, especificando el responsable (si ya ha sido asignado) y el estado de la solicitud (Terminado o Pendiente) 5. Ejecutar la acción correspondiente (crear solicitud o calificar solicitud) 6. Mensaje de Error informando que los datos están incompletos. 7. Mensaje informando que es obligatorio registrar una observación para una calificación de este tipo. 	<ol style="list-style-type: none"> 1. Mostrar las opciones para crear una nueva solicitud. 2. Mostrar las opciones para calificar una solicitud que ya está finalizada. 3. Mostrar la lista de solicitudes que puede eliminar, de las cuales el usuario podrá seleccionar una y esta será eliminada. 4. Mostrar un listado de las ultimas solicitudes realizadas, especificando el responsable (si ya ha sido asignado) y el estado de la solicitud (Terminado o Pendiente) 5. Ejecutar la acción correspondiente (crear solicitud o calificar solicitud) 6. Mensaje de Error informando que los datos están incompletos. 7. Mensaje informando que es obligatorio registrar una observación para una calificación de este tipo.
Calificación de la Prueba: APROBADA ✓	

3.7.4 Prueba de aceptación: HU4 - reserva de equipos.

Tabla 25. Prueba de aceptación: HU4 - reserva de equipos

Prueba de Aceptación – HU4 - RESERVA DE EQUIPOS	
Numero: 4	Historia de Usuario No. 4
<p>Descripción: Con esta prueba se constatará el correcto funcionamiento de la sección de reserva de equipos del sistema de información integrado de la UIT, con el fin de encontrar y resolver errores que se puedan presentar.</p>	
<p>Condiciones de Ejecución: El usuario podrá acceder a estas opciones desde el menú principal, seleccionando la opción “Reserva de Equipos” luego debe seleccionar el tipo de equipos que desea reservar; si son equipos para el acceso a internet, u otro tipo de equipos como video beams, micrófonos, parlantes, cables de sonido, etc.</p>	
<p>Entradas / Pasos de Ejecución: Entradas</p> <ol style="list-style-type: none"> 1. Selecciona “Reservar Equipos para Internet” 2. Selecciona “Reservar Otros Equipos” 3. En la sección “Reserva Equipos para Internet” selecciona una de las opciones (Estudiantes o Docentes) 4. En la sección de “Préstamo de otros equipos”; al usuario le han sido entregados a manera de préstamo, equipos pertenecientes a la UIT 5. Selecciona la opción “Eliminar” en la lista de reservas que el usuario posee 6. En la sección “Reservar otros Equipos”, se cambia la hora en la que desea reservar el recurso 7. El usuario selecciona “Buscar Equipo” 8. El usuario selecciona la opción “Reservar” en la lista de equipos que se muestran disponibles 9. La fecha y la hora de reserva del equipo se vence 10. Acceder al sistema teniendo prestamos cuya devolución ya fue realizada 11. Seleccionar Calificar en la lista de préstamos de equipos que ya fueron devueltos 12. En la lista de opciones de calificación, seleccionar una opción negativa y sin registra observaciones hace clic en “Registrar Calificación” 13. Clic en “Registrar Calificación” 	
<p>Resultados Esperados:</p> <ol style="list-style-type: none"> 1. Se re-direcciona a una página donde tendrá la opción de escoger si es Estudiante o Docente 2. Se re-direcciona a una página donde podrá hacer login para ingresar al sistema de reserva de otros equipos 3. Se re-direcciona a una página donde el usuario podrá hacer login con código y clave de matrícula a los estudiantes o cedula y clave de nómina a los docentes 4. Se muestra un mensaje informando que no está habilitado para realizar 	<p>Resultados Obtenidos:</p> <ol style="list-style-type: none"> 1. Se re-direcciona a una página donde tendrá la opción de escoger si es Estudiante o Docente 2. Se re-direcciona a una página donde podrá hacer login para ingresar al sistema de reserva de otros equipos 3. Se re-direcciona a una página donde el usuario podrá hacer login con su código y clave de matrícula a los estudiantes o cedula y clave de nómina a los docentes 4. Se muestra un mensaje informando que no está habilitado para realizar

Continuación Tabla 25.

Prueba de Aceptación – HU4 - RESERVA DE EQUIPOS	
Numero: 4	Historia de Usuario No. 4
<p>reservas por tener equipos pertenecientes a la UIT en estado de préstamo, también se muestra la lista con los dispositivos cuya devolución está pendiente</p> <p>5. Se realiza el proceso correspondiente a la eliminación de la reserva existente y desaparece la lista</p> <p>6. Se cambia el número de Horas que se puede reservar el recurso, de tal manera que no exceda el horario de 8:00 pm</p> <p>7. Se muestra una lista de equipos que están disponibles para los parámetros de búsqueda seleccionados</p> <p>8. Se guarda la reserva con todos los datos necesarios y actualiza la lista de reservas</p> <p>9. La reserva ya no aparece en la lista de reservas pendientes propia del usuario</p> <p>10. Se muestra una lista donde aparecen los detalles de los préstamos y una opción para “Calificar” en cada uno de los prestamos</p> <p>11. Se muestra una lista con las posibles calificaciones que se pueden dar al préstamo, también aparece un campo donde el usuario puede diligenciar alguna observación que él crea conveniente</p> <p>12. Mensaje informando que es obligatorio registrar una observación para una calificación de este tipo.</p> <p>13. Se guarda la calificación y las observaciones hechas por el usuario para el respectivo préstamo</p>	<p>reservas por tener equipos pertenecientes a la UIT en estado de préstamo, también se muestra la lista con los dispositivos cuya devolución está pendiente</p> <p>5. Se realiza el proceso correspondiente a la eliminación de la reserva existente y desaparece la lista</p> <p>6. Se cambia el número de Horas que se puede reservar el recurso, de tal manera que no exceda el horario de 8:00 pm</p> <p>7. Se muestra una lista de equipos que están disponibles para los parámetros de búsqueda seleccionados</p> <p>8. Se guarda la reserva con todos los datos necesarios y actualiza la lista de reservas</p> <p>9. La reserva ya no aparece en la lista de reservas pendientes propia del usuario</p> <p>10. Se muestra una lista donde aparecen los detalles de los préstamos y una opción para “Calificar” en cada uno de los prestamos</p> <p>11. Se muestra una lista con las posibles calificaciones que se pueden dar al préstamo, también aparece un campo donde el usuario puede diligenciar alguna observación que él crea conveniente.</p> <p>12. Mensaje informando que es obligatorio registrar una observación para una calificación de este tipo.</p> <p>13. Se guarda la calificación y las observaciones hechas por el usuario para el respectivo préstamo</p>
<p>Calificación de la Prueba: APROBADA ✓</p>	

3.7.5 Prueba de aceptación: Hu5 - inventario de equipos.

Tabla 26. Prueba de aceptación: HU5 - inventario de equipos

Prueba de aceptación – HU5 - INVENTARIO DE EQUIPOS	
Numero: 5	Historia de Usuario No. 5
<p>Descripción: Esta prueba está diseñada con el fin de evaluar el funcionamiento del módulo de inventarios del sistema integrado de la UIT, de acuerdo a las especificaciones propuestas en la historia de usuario 5.</p>	
<p>Condiciones de Ejecución: El usuario debe haber superado la autenticación y también debe estar registrado en el sistema UIT, luego deberá acceder al menú Inventario, donde encontrara las opciones que se manejan en este módulo del sistema.</p>	
<p>Entradas / Pasos de Ejecución: Entradas</p> <ol style="list-style-type: none"> 1. Seleccionar crear equipo 2. Seleccionar Crear Lote de Equipos 3. Dar de baja 4. Editar Equipo 5. Entrar/Retirar Equipo Mantenimiento 6. Campos diligenciados correctamente 7. Campos obligatorios están vacíos 8. Agregar/Quitar Dispositivos 9. Dato que se desea ingresar ya existe en la base de datos 10. Dato que se desea ingresar no existe en la base de datos 	
<p>Resultados Esperados:</p> <ol style="list-style-type: none"> 1. Re-direccionar a la página donde se muestran las opciones para guardar todos los datos que se tiene de un equipo. 2. Re-direccionar a la página donde se establece el número de equipos que tendrá el lote, para luego pasar a la página donde se muestran las opciones para guardar todos los datos que se tiene de un equipo. 3. Re-direccionar a la página donde se muestran las opciones donde se puede buscar el equipo a dar de baja. 4. Re-direccionar a la página donde se puede buscar el equipo a editar, para que luego se carguen los datos del mismo en los diferentes controles y puedan ser editados. 	<p>Resultados Obtenidos:</p> <ol style="list-style-type: none"> 1. Re-direccionar a la página donde se muestran las opciones para guardar todos los datos que se tiene de un equipo. 2. Re-direccionar a la página donde se establece el número de equipos que tendrá el lote, para luego pasar a la página donde se muestran las opciones para guardar todos los datos que se tiene de un equipo. 3. Re-direccionar a la página donde se muestran las opciones donde se puede buscar el equipo a dar de baja. 4. Re-direccionar a la página donde se puede buscar el equipo a editar, para que luego se carguen los datos del mismo en los diferentes controles y puedan ser editados.

Continuación Tabla 26.

Prueba de aceptación – HU5 - INVENTARIO DE EQUIPOS	
Numero: 5	Historia de Usuario No. 5
<p>5. Re-direccionar a la página donde se puede buscar el equipo que se desea poner o quitar de mantenimiento, con la opción que permita realizar dicha acción.</p> <p>6. La información registrada se guardara en la base de datos.</p> <p>7. Mensaje informativo donde se explique que los campos son obligatorios.</p> <p>8. Se Mostraran las opciones para agregar la información de los dispositivos extras, si por el contrario lo que se quiere es quitar uno de los dispositivos agregados al equipo, deberá haber la opción para llevar a cabo esta acción.</p> <p>9. Si los datos que se desean registrar ya fueron registrados en el sistema, estos se mostraran en una lista desplegable para evitar la redundancia de información.</p> <p>10. Si los datos que se desean registrar aún no han sido registrados en el sistema, se seleccionara la opción “otra” en la lista desplegable y luego se mostrara una caja de texto donde se podrá registrar la información deseada.</p>	<p>5. Re-direccionar a la página donde se puede buscar el equipo que se desea poner o quitar de mantenimiento, con la opción que permita realizar dicha acción.</p> <p>6. La información registrada se guardara en la base de datos.</p> <p>7. Mensaje informativo donde se explique que los campos son obligatorios.</p> <p>8. Se Mostraran las opciones para agregar la información de los dispositivos extras, si por el contrario lo que se quiere es quitar uno de los dispositivos agregados al equipo, deberá haber la opción para llevar a cabo esta acción.</p> <p>9. Si los datos que se desean registrar ya fueron registrados en el sistema, estos se mostraran en una lista desplegable para evitar la redundancia de información.</p> <p>10. Si los datos que se desean registrar aún no han sido registrados en el sistema, se seleccionara la opción “otra” en la lista desplegable y luego se mostrara una caja de texto donde se podrá registrar la información deseada.</p>
Calificación de la Prueba: APROBADA ✓	

3.7.6 Prueba de aceptación: HU6 – mantenimientos.

Tabla 27. Prueba de aceptación: HU6 – mantenimientos

Prueba de aceptación – HU6 - MANTENIMIENTOS	
Numero: 6	Historia de Usuario No. 6
Descripción: En esta prueba se realizaran los test necesarios para determinar si se implementaron de la manera adecuada las funcionalidades descritas en la historia de usuario donde se establecieron los lineamientos para el manejo de los mantenimientos de los equipos.	
Condiciones de Ejecución: El usuario debe estar trabajando una asistencia o editando un equipo para poder registrar un mantenimiento.	
Entradas / Pasos de Ejecución: Entradas	
<ol style="list-style-type: none"> 1. Hacer clic en Consultar mantenimientos 2. Hacer clic en Agregar Mantenimientos 3. Hacer clic en Anular Mantenimiento 4. Diligenciar campos correctamente 5. Dejar descripción de mantenimiento vacía 	
Resultados Esperados:	Resultados Obtenidos:
<ol style="list-style-type: none"> 1. Se mostrara un historial de los mantenimientos realizados al equipo en cuestión. 2. Se mostraran los controles donde se deberá registrar la información correspondiente al mantenimiento realizado. 3. Se oculta los controles de mantenimiento. 4. Se guarda la información registrada en la base de datos. 5. Se mostrara una advertencia diciendo que la caja de la descripción del mantenimiento está vacía. 	<ol style="list-style-type: none"> 1. Se mostrara un historial de los mantenimientos realizados al equipo en cuestión. 2. Se mostraran los controles donde se deberá registrar la información correspondiente al mantenimiento realizado. 3. Se oculta los controles de mantenimiento. 4. Se guarda la información registrada en la base de datos. 5. Se mostrara una advertencia diciendo que la caja de la descripción del mantenimiento está vacía.
Calificación de la Prueba: APROBADA ✓	

3.7.7 Prueba de aceptación: HU7 – solicitudes.

Tabla 28. Prueba de aceptación: HU7 – solicitudes

Prueba de aceptación – HU7 - SOLICITUDES	
Numero: 7	Historia de Usuario No. 7
<p>Descripción: Se comprobara la integridad de las funcionalidades implementadas en el módulo de asistencias, y si las mismas están acordes con la historia de usuario que las describe.</p>	
<p>Condiciones de Ejecución: Una vez superada la autenticación por parte de un usuario registrado en el sistema integrado de la UIT, deberá acceder a la opción de Solicitudes en el menú principal, posterior a esto, se mostraran las opciones propias para trabajar las necesidades de este módulo.</p>	
<p>Entradas / Pasos de Ejecución: Entradas</p> <ol style="list-style-type: none"> 1. Seleccionar Crear Solicitud. 2. Seleccionar Asignar Responsable a Solicitud. 3. Seleccionar Auto-Asignar Solicitud. 4. Consultar solicitudes Asignadas. 5. Datos diligenciados correctamente. 6. Caja de correo y/o Descripción vacías. 7. No hay monitor seleccionado para asignarlo a una asistencia. 8. Al ver las solicitudes asignadas, no ha seleccionado una para desarrollar. 9. Solicitud seleccionada y hace clic en Ver Equipos. 10. Solicitud seleccionada y hace clic en Finalizar. 11. Con el listado de equipos, hace clic en Vincular y/o Editar 12. Guarda sin registrar mantenimiento 	
<p>Resultados Esperados:</p> <ol style="list-style-type: none"> 1. Re-direccionara a una página donde los usuarios registrados en el sistema integrado UIT podrán crear solicitudes de docentes y funcionarios de la universidad. 2. Re-direccionar a la página donde se listaran todas las asistencias que aún no tiene responsable, luego se mostraran las opciones para seleccionar uno de los monitores que se encargara de dar solución al problema referente. 3. Esta opción remitirá al usuario a una página donde se listan las solicitudes sin asignar un responsable, de las cuelas el usuario podrá seleccionar la 	<p>Resultados Obtenidos:</p> <ol style="list-style-type: none"> 1. Re-direccionara a una página donde los usuarios registrados en el sistema integrado UIT podrán crear solicitudes de docentes y funcionarios de la universidad. 2. Re-direccionar a la página donde se listaran todas las asistencias que aún no tiene responsable, luego se mostraran las opciones para seleccionar uno de los monitores que se encargara de dar solución al problema referente. 3. Esta opción remitirá al usuario a una página donde se listan las solicitudes sin asignar un responsable, de las cuelas el usuario podrá seleccionar la que desea auto-asignarse.

Continuación Tabla 28

Prueba de aceptación – HU7 - SOLICITUDES	
Numero: 7	Historia de Usuario No. 7
<p>que desea auto-asignarse.</p> <p>4. Se remitirá al usuario a la página donde se listarán las solicitudes que le han sido asignadas, para que las desarrolle.</p> <p>5. La información registrada se guardará permanentemente en la base de datos.</p> <p>6. Alerta indicando que los datos están incompletos.</p> <p>7. Deshabilitar los botones que generan acciones.</p> <p>8. Deshabilitar los botones que generan acciones.</p> <p>9. Se listarán los equipos que existen en la dependencia.</p> <p>10. La solicitud se cambiara de estado, pasando de pendiente a finalizada, y ya no se listara en las opcionales para ser trabajada.</p> <p>11. Se cargara la información del equipo en los diferentes controles, para que el usuario pueda editar las partes y también registrar el mantenimiento realizado al equipo, una vez realizado todo este proceso, se guardaran los cambios en la Base de datos.</p> <p>12. Mensaje de error informando que los mantenimientos cuando se trabaja una asistencia son obligatorios.</p>	<p>4. Se remitirá al usuario a la página donde se listarán las solicitudes que le han sido asignadas, para que las desarrolle.</p> <p>5. La información registrada se guardará permanentemente en la base de datos.</p> <p>6. Alerta indicando que los datos están incompletos.</p> <p>7. Deshabilitar los botones que generan acciones.</p> <p>8. Deshabilitar los botones que generan acciones.</p> <p>9. Se listarán los equipos que existen en la dependencia.</p> <p>10. La solicitud se cambiara de estado, pasando de pendiente a finalizada, y ya no se listara en las opcionales para ser trabajada.</p> <p>11. Se cargara la información del equipo en los diferentes controles, para que el usuario pueda editar las partes y también registrar el mantenimiento realizado al equipo, una vez realizado todo este proceso, se guardaran los cambios en la Base de datos.</p> <p>12. Mensaje de error informando que los mantenimientos cuando se trabaja una asistencia son obligatorios.</p>
Calificación de la Prueba: APROBADA ✓	

3.7.8 Prueba de aceptación: HU8 - horarios monitores.

Tabla 29. Prueba de aceptación: HU8 - horarios monitores

Prueba de aceptación – HU8 - HORARIOS MONITORES	
Numero: 8	Historia de Usuario No. 8
<p>Descripción: El test está diseñado para evaluar el buen funcionamiento del módulo Horarios Monitores en el sistema integrado UIT. Con el fin de encontrar falencias y corregirlas de la mejor manera posible.</p>	
<p>Condiciones de Ejecución: El usuario debe haber hecho login en el sistema, y en el menú principal deberá seleccionar la opción Horarios M. posterior a esto, se mostraran todas las opciones disponibles para que el usuario autorizado pueda crear, editar, eliminar, consular e imprimir los horarios con los cuales se controlan las jornadas de trabajo de los monitores que laboran en la UIT.</p>	
<p>Entradas / Pasos de Ejecución: Entradas</p> <ol style="list-style-type: none"> 1. Seleccionar la opción Crear Nuevo Horario. 2. Seleccionar la opción Editar Horario Vigente. 3. Seleccionar la opción Eliminar Horario. 4. Seleccionar la opción Consultar Horarios Vigentes. 5. Nombre nuevo horario está vacío. 6. Usuario completa las horas establecidas el tipo de monitoria. 7. Usuario ya tiene horas en otro horario. 8. No hay horario para Editar 9. No hay horario para Eliminar 10. No hay horarios para consultar Horarios Vigentes. 	
<p>Resultados Esperados:</p> <ol style="list-style-type: none"> 1. Re-direccionar a la página donde se podrá escoger el nombre del nuevo horario, y luego pasara a una página donde podrá seleccionar a uno de los monitores y sus respectivas horas laborales. 2. Re-direccionar a la página donde se podrá el horario a editar de una lista desplegable, y luego pasara a una página donde podrá seleccionar a uno de los monitores y sus respectivas horas laborales. 3. Re-direccionar a la página donde se podrá el horario a eliminar de una lista desplegable, y luego se realizara dicha acción presionado el botón Eliminar. 4. Re-direccionar a la página donde se 	<p>Resultados Obtenidos:</p> <ol style="list-style-type: none"> 1. Re-direccionar a la página donde se podrá escoger el nombre del nuevo horario, y luego pasara a una página donde podrá seleccionar a uno de los monitores y sus respectivas horas laborales. 2. Re-direccionar a la página donde se podrá el horario a editar de una lista desplegable, y luego pasara a una página donde podrá seleccionar a uno de los monitores y sus respectivas horas laborales. 3. Re-direccionar a la página donde se podrá el horario a eliminar de una lista desplegable, y luego se realizara dicha acción presionado el botón Eliminar. 4. Re-direccionar a la página donde se

Continuación Tabla 29.

Prueba de aceptación – HU8 - HORARIOS MONITORES	
Numero: 8	Historia de Usuario No. 8
<p>podrá visualizar las versiones finales de los horarios existentes en el sistema, según sea la selección del usuario, también permitirá imprimir el horario que se esté mostrando.</p> <p>5. Se muestra un mensaje de alerta en el que se le informara al usuario que los datos están incompletos.</p> <p>6. Se deshabilita las otras opciones de selección, para que no sea posible marcar más horas de las establecidas.</p> <p>7. Las opciones de selección que corresponden al día y la hora que ya ha sido seleccionada en otro horario no estarán visibles para el usuario.</p> <p>8. Se deshabilitan los controles que ejecutan acciones.</p> <p>9. Se deshabilitan los controles que ejecutan acciones.</p> <p>10. El botón imprimir no se muestra al usuario.</p>	<p>podrá visualizar las versiones finales de los horarios existentes en el sistema, según sea la selección del usuario, también permitirá imprimir el horario que se esté mostrando.</p> <p>5. Se muestra un mensaje de alerta en el que se le informara al usuario que los datos están incompletos.</p> <p>6. Se deshabilita las otras opciones de selección, para que no sea posible marcar más horas de las establecidas.</p> <p>7. Las opciones de selección que corresponden al día y la hora que ya ha sido seleccionada en otro horario no estarán visibles para el usuario.</p> <p>8. Se deshabilitan los controles que ejecutan acciones.</p> <p>9. Se deshabilitan los controles que ejecutan acciones.</p> <p>10. El botón imprimir no se muestra al usuario.</p>
<p>Calificación de la Prueba: APROBADA ✓</p>	

3.7.9 Prueba de aceptación: HU9 – aulas.

Tabla 30. Prueba de aceptación: HU9 – aulas

Prueba de aceptación – HU9 - AULAS	
Numero: 9	Historia de Usuario No.9
Descripción: Prueba que evalúa las funcionalidades del módulo de administración de aulas del sistema de información integrado de la UIT, la finalidad es evitar errores de funcionamiento que afecten el funcionamiento del mismo	
Condiciones de Ejecución: Una vez superada la parte de autenticación al sistema de información de la UIT, se deberá seleccionar la opción “Aulas”.	
Entradas / Pasos de Ejecución: Entradas	
<ol style="list-style-type: none"> 1. Seleccionar la opción Administrar Aulas 2. Selecciona la opción Registrar SW instalado en las aulas 3. Selecciona la opción Administrar Cursos 4. Selecciona la opción Ver Horario Final Aula 5. Cajas de texto vacías 6. Listas desplegadas sin selección 7. Clic en guardar de la sección Crear Aula 8. Clic en guardar de la sección Editar Aula 9. Clic en Eliminar de la sección Eliminar Aula 10. Clic en el Botón agregar de la sección Registrar Software Instalado en las Aulas 11. Clic en Quitar de la sección Registrar Software Instalado en las Aulas 12. Clic en el botón Buscar Horarios de la sección Administrar Cursos 13. Clic en el botón Cambiar Parámetros de Búsqueda de la sección Administrar Cursos 14. Clic seleccionando uno de los horarios disponibles. 15. Clic en Eliminar curso de la sección Administrar Cursos 16. Clic en Imprimir en la sección Ver Horario Final Aula 	
Resultados Esperados:	Resultados Obtenidos:
<ol style="list-style-type: none"> 1. El usuario se re-direcciona a la sección donde podrá crear, editar y eliminar las diferentes aulas que existen en la Udenar. 2. El usuario se re-direcciona a la sección donde podrá agregar o quitar Software que está instalado en las diferentes aulas que existen en la Institución. 3. El usuario se re-direcciona a la sección donde podrá crear y eliminar los diferentes cursos que se dictan en las aulas de informática que están a cargo de la UIT, así como fijar su horario y las instalaciones físicas donde se impartirá 	<ol style="list-style-type: none"> 1. El usuario se re-direcciona a la sección donde podrá crear, editar y eliminar las diferentes aulas que existen en la Udenar. 2. El usuario se re-direcciona a la sección donde podrá agregar o quitar Software que está instalado en las diferentes aulas que existen en la Institución. 3. El usuario se re-direcciona a la sección donde podrá crear y eliminar los diferentes cursos que se dictan en las aulas de informática que están a cargo de la UIT, así como fijar su horario y las instalaciones físicas donde se impartirá

Continuación Tabla 30.

Prueba de aceptación – HU9 - AULAS	
Numero: 9	Historia de Usuario No.9
<p>dicha clase.</p> <p>4. Al usuario se le mostrara una página donde se mostrara el horario completo de clases de un aula determinada, también tendrá la opción de imprimir dicho horario.</p> <p>5. Mensaje informando que los datos están incompletos</p> <p>6. Mensaje informativo mostrando que debe seleccionarse un ítem.</p> <p>7. Deberá Guardar en la Base de datos la información de la nueva aula.</p> <p>8. Deberá Editar en la Base de datos la nueva información del aula existente seleccionada.</p> <p>9. Deberá Eliminar de la Base de datos información del aula existente seleccionada.</p> <p>10. Se agregara los datos registrados al Software instalado en el aula de informática seleccionada</p> <p>11. Se quitara el ítem seleccionado del software que está instalado en el aula seleccionada</p> <p>12. Cuando se pulsa esta opción, se deshabilitan todos los controles que interfieren para dicha búsqueda y se muestran los horarios opcionales para ubicar el curso.</p> <p>13. Se volverán a habilitar los controles que intervienen como parámetros de búsqueda.</p> <p>14. Se guarda el curso, con toda la información pertinente y en el horario que se selecciona.</p> <p>15. Eliminará el curso seleccionado de la base de datos.</p> <p>16. Se abrirá el menú de impresión del horario del aula seleccionada previamente</p>	<p>dicha clase.</p> <p>4. Al usuario se le mostrara una página donde se mostrara el horario completo de clases de un aula determinada, también tendrá la opción de imprimir dicho horario.</p> <p>5. Mensaje informando que los datos están incompletos</p> <p>6. Mensaje informativo mostrando que debe seleccionarse un ítem.</p> <p>7. Deberá Guardar en la Base de datos la información de la nueva aula.</p> <p>8. Deberá Editar en la Base de datos la nueva información del aula existente seleccionada.</p> <p>9. Deberá Eliminar de la Base de datos información del aula existente seleccionada.</p> <p>10. Se agregara los datos registrados al Software instalado en el aula de informática seleccionada</p> <p>11. Se quitara el ítem seleccionado del software que está instalado en el aula seleccionada</p> <p>12. Cuando se pulsa esta opción, se deshabilitan todos los controles que interfieren para dicha búsqueda y se muestran los horarios opcionales para ubicar el curso.</p> <p>13. Se volverán a habilitar los controles que intervienen como parámetros de búsqueda.</p> <p>14. Se guarda el curso, con toda la información pertinente y en el horario que se selecciona.</p> <p>15. Eliminará el curso seleccionado de la base de datos.</p> <p>16. Se abrirá el menú de impresión del horario del aula seleccionada previamente</p>
Calificación de la Prueba: APROBADA ✓	

3.7.10 Prueba de aceptación: HU10 – prestamos.

Tabla 31. Prueba de aceptación: HU10 – prestamos

Prueba de aceptación – HU10 - PRESTAMOS	
Numero: 10	Historia de Usuario No. 10
Descripción: Esta prueba se realiza con el fin de determinar que todas las funciones de la sección Préstamos del sistema integrado de la UIT no presenten errores y cumplan con los requerimientos solicitados por el cliente.	
Condiciones de Ejecución: Se debe haber superado la autenticación principal del sistema y acceder a la sección préstamos.	
Entradas / Pasos de Ejecución: Entradas	
<ol style="list-style-type: none"> 1. Hacer clic en “Ver Recursos Prestamos” 2. Hacer clic en “Administrar Prestamos Otros Equipos” 3. Hacer clic en “Buscar Todos los Recursos” 4. Hacer clic en “Buscar Recursos Habilitados” 5. Hacer clic en “Buscar Recursos en Mantenimiento” 6. Después de seleccionar una de las 3 opciones anteriores, hacer clic en “Quitar” en uno de los equipos listados 7. En la opción “Administrar prestamos otros equipos” seleccionar “Entregar Equipo” 8. Hacer clic en “Entregar” 9. Hacer clic en “Imprimir” 10. En la opción “Administrar prestamos otros equipos” seleccionar “Devolver Equipo” 11. Seleccionar la opción “Devolver” 12. En la sección “Devolver Equipo” seleccionar “Guardar” 	
Resultados Esperados:	Resultados Obtenidos:
<ol style="list-style-type: none"> 1. Se re-direcciona a una página donde se muestran las opciones para poder visualizar los recursos disponibles para prestamos 2. Se re-direcciona a una página donde están un login para ingresar a la sección de administración de préstamos de otros equipos 3. Se listan todos los recursos que han sido habilitados para préstamo al publico 4. Se listan los recursos que han sido habilitados para préstamo al público y que no estén en mantenimiento o dados de baja 	<ol style="list-style-type: none"> 1. Se re-direcciona a una página donde se muestran las opciones para poder visualizar los recursos disponibles para prestamos 2. Se re-direcciona a una página donde están un login para ingresar a la sección de administración de préstamos de otros equipos 3. Se listan todos los recursos que han sido habilitados para préstamo al publico 4. Se listan los recursos que han sido habilitados para préstamo al público y que no estén en mantenimiento o dados de baja

Continuación Tabla 31.

Prueba de aceptación – HU10 - PRESTAMOS	
Numero: 10	Historia de Usuario No.10
<p>5. Se listan los recursos que han sido habilitados para préstamo al público y que estén en mantenimiento</p> <p>6. Se desactiva el estado de préstamo al equipo seleccionado para que no aparezca disponible y se refresca la lista</p> <p>7. Se muestra una lista de los equipos que han sido reservados y sus atributos pertinentes, cada uno tendrá la opción “Entregar”</p> <p>8. Se cambia el estado del préstamo, pasando de “Reservado” a “Entregado” y se muestra un formato imprimible con la información correspondiente</p> <p>9. Se muestran las opciones para realizar la impresión del formato creado para el préstamo</p> <p>10. Se muestra una lista de los equipos cuyo estado de préstamo es “Entregado” con sus atributos pertinentes, cada uno tendrá la opción “Devolver”</p> <p>11. Se muestra un campo donde el funcionario podrá registrar alguna observación o anomalía del equipo en el momento de la devolución</p> <p>12. Se cambia el estado del préstamo, pasando de “Entregado” a “Devuelto” y se actualiza la lista de equipos entregados</p>	<p>5. Se listan los recursos que han sido habilitados para préstamo al público y que estén en mantenimiento</p> <p>6. Se desactiva el estado de préstamo al equipo seleccionado para que no aparezca disponible y se refresca la lista</p> <p>7. Se muestra una lista de los equipos que han sido reservados y sus atributos pertinentes, cada uno tendrá la opción “Entregar”</p> <p>8. Se cambia el estado del préstamo, pasando de “Reservado” a “Entregado” y se muestra un formato imprimible con la información correspondiente</p> <p>9. Se muestran las opciones para realizar la impresión del formato creado para el préstamo</p> <p>10. Se muestra una lista de los equipos cuyo estado de préstamo es “Entregado” con sus atributos pertinentes, cada uno tendrá la opción “Devolver”</p> <p>11. Se muestra un campo donde el funcionario podrá registrar alguna observación o anomalía del equipo en el momento de la devolución</p> <p>12. Se cambia el estado del préstamo, pasando de “Entregado” a “Devuelto” y se actualiza la lista de equipos entregados</p>
<p>Calificación de la Prueba: APROBADA ✓</p>	

3.7.11 Prueba de aceptación: HU11 – reportes.

Tabla 32. Prueba de aceptación: HU11 - REPORTES

Prueba de aceptación – HU11 - REPORTES	
Numero:11	Historia de Usuario No.11
<p>Descripción: Esta prueba evalúa el funcionamiento del módulo de reportes del sistema integrado de la UIT, con el fin de que no se presenten errores de funcionamiento y validación.</p>	
<p>Condiciones de Ejecución: El usuario que tenga autorización para realizar reportes, tendrá que haber superado la autenticación del sistema y posteriormente dirigirse a la opción Reportes que se encuentra en el menú principal.</p>	
<p>Entradas / Pasos de Ejecución: Entradas</p> <ol style="list-style-type: none"> 1. Seleccionar la opción Reporte Asistencias 2. Seleccionar la opción Reporte Mantenimientos 3. Seleccionar la opción Reporte Asistencias por Dependencias 4. Cambiar alguna de las fechas en el formulario para generar reportes de Mantenimientos 5. Clic en el botón Generar Reporte 6. Clic en la Opción Exportar Reporte 7. Clic en una de las opciones del Menú Exportar Reportes 8. Clic en la opción Imprimir Reporte 	
<p>Resultados Esperados:</p> <ol style="list-style-type: none"> 1. El usuario se re-direcciona a una página donde puede crear el reporte de asistencias solicitadas a la UIT 2. El usuario se re-direcciona a una página donde puede crear el reporte de Mantenimientos realizados por los funcionarios de la UIT 3. El usuario se re-direcciona a una página donde puede crear el reporte de asistencias solicitadas a la UIT agrupadas por dependencias, por calificación y por estado 4. Se actualizan las listas de selección de usuarios y clasificación con las opciones específicas de dicho periodo 5. Se generara el reporte de acuerdo a las fechas de inicio y fin seleccionadas, en el caso del reporte de mantenimientos, también se tendrán en cuenta los parámetros de Tipo de 	<p>Resultados Obtenidos:</p> <ol style="list-style-type: none"> 1. El usuario se re-direcciona a una página donde puede crear el reporte de asistencias solicitadas a la UIT 2. El usuario se re-direcciona a una página donde puede crear el reporte de Mantenimientos realizados por los funcionarios de la UIT 3. El usuario se re-direcciona a una página donde puede crear el reporte de asistencias solicitadas a la UIT agrupadas por dependencias, por calificación y por estado 4. Se actualizan las listas de selección de usuarios y clasificación con las opciones específicas de dicho periodo 5. Se generara el reporte de acuerdo a las fechas de inicio y fin seleccionadas, en el caso del reporte de mantenimientos, también se tendrán en cuenta los parámetros de Tipo de

Continuación Tabla 32.

Prueba de aceptación – HU11 - REPORTES	
Numero:11	Historia de Usuario No.11
<p>Mantenimiento, Responsable y Calificación</p> <p>6. Se muestra las opciones de Exportar la información en formato PDF, Excel o Word</p> <p>7. Se solicita al usuario la ubicación y el nombre del archivo donde se guardarán los datos exportados</p> <p>8. Se muestran las opciones propias del procedimiento imprimir, de acuerdo al sistema utilizado para la visualización del reporte</p>	<p>Mantenimiento, Responsable y Calificación</p> <p>6. Se muestra las opciones de Exportar la información en formato PDF, Excel o Word</p> <p>7. Se solicita al usuario la ubicación y el nombre del archivo donde se guardarán los datos exportados</p> <p>8. Se muestran las opciones propias del procedimiento imprimir, de acuerdo al sistema utilizado para la visualización del reporte</p>
Calificación de la Prueba: APROBADA ✓	

4. CONCLUSIONES

La programación extrema es una metodología robusta y confiable, con la que se puede generar sistemas de información de gran dimensión y satisfaciendo las necesidades del cliente por completo.

La metodología XP genera aplicaciones escalables y fáciles de mantener en el tiempo, además, genera código fácil de entender para otros desarrolladores debido a las normativas de comentarios y definición de variables altamente descriptivas.

Las herramientas de desarrollo, despliegue y almacenamiento de datos utilizados, garantizan un adecuado nivel de seguridad, alto grado de persistencia de la información almacenada en la base de datos, también la estabilidad del sistema, entre muchas otras ventajas propias del alto acoplamiento de este tipo de sistemas.

Un sistema integrado multiplica los beneficios, en comparación con varios sistemas que manejan los procesos por separado, además de elevar el nivel de organización de información manejada por la organización.

El sistema integrado de la UIT eleva la productividad de la dependencia, ya que agiliza las tareas a realizar por parte de los directivos, funcionarios y monitores de que se encuentran vinculados a la misma.

El proyecto actual permite aplicar los conocimientos que fueron adquiridos durante el transcurso de la carrera en Ingeniería de Sistemas de la Universidad de Nariño y permiten que el estudiante obtenga experiencia tanto en la práctica de sus conocimientos como a nivel laboral gracias a la modalidad Pasantía.

5. RECOMENDACIONES

El sistema de información que se ha desarrollado tiene inmensas opciones de mejorar, complementar y optimizar, puesto que no era posible abarcar la totalidad de las necesidades de la UIT dentro de este proyecto, teniendo en cuenta lo anteriormente mencionado, se recomienda tratar de que las partes que sean posteriormente desarrolladas se acoplen de la mejor manera posible con el sistema actual.

Para obtener un buen funcionamiento del sistema desarrollado, se recomienda:

Desplegar en un equipo cuyos recursos tanto de hardware como de software sean relativamente actuales.

Brindar una adecuada conexión a Internet, con una dirección IP pública propia y/o un nombre de dominio institucional adecuado.

Realizar backups de la Base de Datos periódicamente y almacenarlos en un sitio seguro, con el fin de tener un respaldo en una situación de emergencia. Ya que la información que se maneja en el sistema de información es cambiante y toda esta registrada en la base de datos de SQL server

Tener cuidado con las personas que tiene acceso al módulo de reportes, ya que es información importante, y por ende debe ser manejada con confidencialidad.

BIBLIOGRAFÍA

BACHRACH, Ann; KOMO, Darmadi; ASHOK, Gopal; HODGINS, Joanne; CHIU, Oliver; MCBRIDE, Sabrena; THARIAN, Tharun y TEJEDOR, Michael. SQL Server 2012, Microsoft Press, Unica Edicion. November 2011, Updated March 2012, 20 p.

CORTIZO PEREZ, Jose; EXPOSITO G, Diego y RUIZ LEYVA, Miguel. eXtreme Programing, 2012. 97p.

Extreme Programming XP, [en línea]. [Fecha de consulta: 16 octubre de 2012]. En: http://www.slideshare.net/joaquin_win/extreme-programming-456979

FOWLER, Martin. Is Design Dead, Martin Fowler [En Línea]. [Fecha de consulta: 22 de febrero de 2014]. Disponible en: <http://martinfowler.com/articles/designDead.html>

LANDA, Nicolas. C#. Guía total del programador. Fox Andina – Lomas de Zamora, primera edición. Buenos Aires 2010, 404 p.

LAURIAT, Shawn M. Advanced Ajax, Pearson Education S.A, 2ªed. Boston, 2008. 360 p.

Microsoft SQL Server. En: Wikipedia [En Línea]. [Fecha de consulta: 04 de febrero de 2014]. Disponible en: http://es.wikipedia.org/wiki/Microsoft_SQL_Server

PELLAND, Patrice; PARE, Pascal y HAINES, ken. Movingto Visual Studio 2010, Microsoft Press, Unica Edicion. Washington 2011, 319 p.

PHP. En: Wikipedia [En Línea]. [Fecha de consulta: 02 de Mayo de 2014]. Disponible en: <http://es.wikipedia.org/wiki/PHP>

Rational Unified Process – RUP, [en línea]. [Fecha de consulta: 15 octubre de 2012]. Disponible en: http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

Refactorización. En: Wikipedia [En Línea]. [Fecha de consulta: 28 de febrero de 2014]. Disponible en: <http://es.wikipedia.org/wiki/Refactorizaci%C3%B3n>

SOMMERVILLE, Ian. "Ingeniería del Software". Pearson Educación S.A, séptima edición. Madrid 2005. 687 p.

The Agile Unified Process– AUP, [en línea]. [Fecha de consulta: 15 octubre de 2012]. Disponible en: <http://www.ambyssoft.com/unifiedprocess/agileUP.html>