



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA DE INGENIERIA EN INFORMÁTICA Y SISTEMAS

COMPUTACIONALES

PROYECTO DE INVESTIGACIÓN

TEMA:

**“DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE
EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI
EXTENSIÓN LA MANÁ”**

Proyecto de titulación previo a la obtención del título de Ingeniero en Informática y Sistemas
Computacionales

Autor:

Chamaidan Panchana Rufino Yimael

Tutor:

Ing. MSc. Chanatasig Toapanta Henry Mauricio

LA MANÁ – ECUADOR

Agosto-2017

DECLARACIÓN DE AUTORÍA

“Yo, Rufino Yimael Chamaidan Panchana declaro ser autor del presente proyecto de investigación **“DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ”**, siendo el Ing. MSc. Henry Chanatasig tutor del presente trabajo; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certificamos que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de nuestra exclusiva responsabilidad.

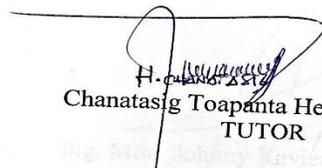


Chamaidan Panchana Rufino Yimael
C.I: 0704006873

AVAL DEL TUTOR DE PROYECTO DE INVESTIGACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título: **“DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ”**, del señor estudiante; Rufino Yimael Chamaidan Panchana de la carrera de Ingeniería en Informática en Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Honorable Consejo Académico de la Unidad Académica de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

La Maná, 21 de Agosto de 2017



H. Chanatasig Toapanta Henry Mauricio
Chanatasig Toapanta Henry Mauricio
TUTOR

APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Unidad Académica de Ciencias de la Ingeniería y Aplicadas; por cuanto, al postulante: del señor estudiante; Rufino Yimael Chamaidan Panchana con el Título de Proyecto de Investigación: **“DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ”**, han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

La Maná, 27 de Agosto 2017

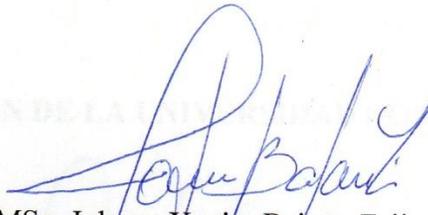
Para constancia firman;



MSc. Jaime Mesías Cajas
C.I: 0502359250
LECTOR 1 PRESIDENTE



Ing. MSc. Diego Fernando Jácome Segovia
C.I: 0502554082
LECTOR 2



Ing. MSc. Johnny Xavier Bazaña Zajia
LECTOR 3 SECRETARIO
C.I: 1204827115



Universidad
Técnica de
Cotopaxi



Trabajo de
Grado
CIYA

COORDINACIÓN
TRABAJO DE GRADO

CERTIFICACIÓN

El suscrito. Lic. Mg. Sc. López Bustamante Ringo John con C.I: 1202797112, **COORDINADOR ACADÉMICO Y ADMINISTRATIVO DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI - EXTENSIÓN LA MANÁ** certifica que:

El Sr. Chamaidan Panchana Rufino Yimael , estudiante de la Universidad Técnica de Cotopaxi extensión La Maná, cumplió a cabalidad con la realización y entrega del proyecto con el nombre **“DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ”**, el mismo que cumple con todos los requerimientos establecidos en el transcurso de su investigación.

Es todo cuanto puedo certificar en honor a la verdad y autorizo al interesado hacer uso del presente documento siempre y cuando este dentro de las leyes.

Particular que comunico para fines pertinentes

Atentamente;

“POR LA VINCULACIÓN DE LA UNIVERSIDAD CON EL PUEBLO”



La Maná, agosto 01 del 2017

Lcdo. Mg. Sc. Ringo López Bustamante
COORDINADOR DE LA EXTENSIÓN
Universidad Técnica de Cotopaxi – La Maná

RLB/eas

DEDICATORIA

Esta investigación la quiero dedicar a mi madre, a mi padre, a mi esposa, y a mi hijo por ser ese pilar fundamental en el propósito alcanzado, por brindarme su apoyo incondicional en los momentos más difíciles y sobre todo a DIOS porque con el todo es posible.

Yimael

AGRADECIMIENTO

Un profundo agradeciendo a mí querida universidad por haberme permitido formarme en sus aulas.

A mis maestros por gran capacidad al impartir sus conocimientos para formarme con un profesional íntegro y con capacidades.

Yimael



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TEMA: “DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ”

Autor: Rufino Yimael Chamaidan Panchana

RESUMEN

Estar conscientes de los avances tecnológicos hace notar que en los últimos años han contribuido sobremedida al desarrollo de la sociedad, es así que gran parte de las instituciones públicas y privadas se empeñan en aplicar nuevas tecnologías que aporten en la mejora sustancial de sus procesos, sin duda alguna la Universidad Técnica de Cotopaxi Extensión La Maná busca mejorar y sistematizar los procesos con el propósito de alcanzar un mayor nivel de eficiencia, de tal manera se propuso desarrollar un sistema informático para la gestión de equipos tecnológicos de la institución que permita mejorar la forma de registro del historial de mantenimiento tanto preventivo como correctivo de los equipos, además que brinde información de la dependencia de ubicación, responsable, componentes internos como externos, la marca y el software de los equipos; el sistema se desarrolló con herramientas específicas para el caso como son: case para el modelado de la base de datos, un motor de base de datos SQLServer, la programación se efectuó con el lenguaje de programación visual.net para el desarrollo de la interfaz gráfica; el sistema proporcionará información completa sobre todos los equipos tecnológicos con que cuenta la Universidad Técnica de Cotopaxi Extensión La Maná, el aporte que esta aplicación presta es la facilidad de obtener reportes sobre toda la información relacionada a los bienes informáticos que están a cargo del departamento de servicios informáticos, otras de las funcionalidades que brinda es la determinación de su vida útil y por ende dar de baja a un equipo que ya cumplieron con su vida útil, la aplicación se desarrollará con la metodología de desarrollo de software Scrum debido que presta todas las guías para el desarrollo de los sistemas.

Palabras claves: *Sistema informático, Herramientas Case, SqlServer, Visula.net, Interfaz Grafica*



TECHNICAL UNIVERSITY OF COTOPAXI

FACULTY OF ENGINEERING SCIENCES AND APPLIED

THEME: "DEVELOPMENT OF A COMPUTER SYSTEM FOR THE MANAGEMENT OF TECHNOLOGICAL EQUIPMENT OF THE TECHNICAL UNIVERSITY OF COTOPAXI EXTENSION LA MANÁ"

Autor: Rufino Yimael Chamaidan Panchana

ABSTRACT

Being aware of the technological advances makes note that in recent years have contributed greatly to the development of society, so much of public and private institutions are committed to apply new technologies that contribute to the substantial improvement of their processes, without Doubt the Technical University of Cotopaxi Extension La Maná seeks to improve and systematize the processes with the purpose of achieving a higher level of efficiency, so it was proposed to develop a computer system for the management of technological equipment of the institution that allows to improve the way Record of both preventative and corrective maintenance history of the equipment, in addition to providing information on the dependency of location, responsible, internal and external components, brand and equipment software; the system was developed with specific tools for the case such as: case for modeling the database, a database engine SQLServer, programming was done with the programming language visual.net for the development of the graphical interface ; the system will provide complete information on all the technological equipment that the Technical University of Cotopaxi Extension La Maná has, the contribution that this application provides is the facility to obtain reports on all information related to the computer goods that are in charge of the department of computer services, other of the functionalities that it provides is the determination of its useful life and therefore to derive to a equipment that already fulfilled with its useful life, the application will be developed with the methodology of development of software Scrum due it renders all the Guides for the development of the systems.

Keywords: Computer System, Case Tools, Sql Server, Visual.net, Graphic Interface

ÍNDICE GENERAL

PORTADA	i
DECLARACIÓN DE AUTORÍA	ii
AVAL DEL TUTOR DE PROYECTO DE INVESTIGACIÓN	iii
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN.....	iv
CERTIFICACIÓN.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
RESUMEN	viii
ABSTRACT.....	ix
ÍNDICE GENERAL.....	x
1. INFORMACIÓN GENERAL	1
2. RESUMEN DEL PROYECTO	2
3. JUSTIFICACIÓN DEL PROYECTO	3
4. BENEFICIARIOS DEL PROYECTO	4
5. PROBLEMA DE INVESTIGACIÓN	4
6. OBJETIVOS.....	6
6.1. Objetivo General	6
6.2. Objetivos Específicos	6
7. SISTEMA DE TAREAS EN RELACIÓN A OBJETIVOS PLANTEADOS	7
8. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA.....	8
8.1. Software y su ciclo de vida.....	8
8.1.1. Modelos de ciclo de vida del software	8
8.1.1.1. El modelo en cascada	8

8.1.1.2.	Análisis del sistema	9
8.2.	Metodología de desarrollo Scrum.....	10
8.2.1.	¿Qué es Scrum?	10
8.3.	Proceso de desarrollo de software	10
8.3.1.	Pasos del proceso de desarrollo de software	11
8.3.2.	Categorías de paradigmas de programación.....	12
8.3.3.	Diagramas de flujos de datos.....	13
8.4.	Sistema de control	13
8.5.	SQL server 2008.....	14
8.5.1.	Características en SQL Server 2008.....	15
8.5.2.	Ventajas SQL Server 2008	15
8.5.3.	Desventajas SQL Server 2008.....	15
8.5.4.	Servidor de Base de datos SQL Server 2008.....	16
8.6.	Visual Studio 2010	16
8.7.	Metodología XP (Extreme Programming).	17
8.7.1.	Descripción de Metodología (XP).....	17
8.7.2.	Características de XP.....	18
8.7.3.	Ventajas	18
8.8.	Sistema cliente servidor.....	19
8.8.1.	Características del modelo cliente/servidor.....	19
8.8.2.	Ventajas del modelo cliente/servidor	19
8.8.3.	Funciones del cliente	20
8.9.	Herramientas para el diseño del sistema de control	20
8.9. 1.	La replicación	20
8.9. 2.	Herramientas CASE	21

8.9.2.1.	Visual Paradigm	21
8.9.2.1.1.	Propósito de Visual Paradigm	21
8.9. 2. 1. 2.	Características de Visual Paradigm	21
8.9.3.	Lenguaje de programación	22
8.9.3. 1.	Sencillez de uso	23
8.9.3. 2.	Compatible	23
8.9.3. 3.	Modernidad.....	23
8.9.3. 4.	Orientado a objetos.....	23
8.9.3. 5.	Recolección de basura	24
8.9.3.6.	Extensión de los operadores básicos	24
8.9.3. 7.	Seguridad de tipos	24
8.9.3.8.	Instrucciones seguras.....	24
8.9.3.9.	Unificación de tipos.....	24
8.9.3.10.	Extensión de los operadores básicos	25
8.9.3. 11.	Eficiente.....	25
8.9.4.	Metodología Scrum	25
8.9.4. 1.	Beneficios	26
8.9.4. 2.	Proceso de Scrum	27
8.9. 5.	Roles de SCRUM	28
8.9.6.	Herramientas de Programación	29
8.9.6.1.	Visual Studio 2010	29
9.	HIPÓTESIS	30
10.	METODOLOGÍA DE LA INVESTIGACIÓN	30
10.1.	Investigación de campo	30
10.2.	Investigación bibliográfica y documental.....	30

10.3.	MÉTODOS DE INVESTIGACIÓN	30
10.3.1.	Método inductivo.....	30
10.3.2.	Método deductivo.....	31
10.4.2.	Encuesta.....	31
10.5.	DISEÑO EXPERIMENTAL.....	32
11.	ANÁLISIS Y DISCUSIÓN DE RESULTADOS	32
11.1.	Muestra	32
11.2.	Análisis de resultados de las encuestas.....	33
11.3.	Requisitos mínimos del sistema	33
11.4.	Requerimientos del sistema	33
11.5.	Requerimientos para el desarrollo del sistema	34
11.5.1.	Requerimientos Funcionales.....	34
11.4.2.	Requerimientos no funcionales	36
11.6.	Funcionalidades del sistema	37
11.7.	Requisitos de Rendimiento.....	38
11.7.1.	Usabilidad.....	38
11.7.2.	Seguridad	38
11.7.3.	Multiplataforma.....	38
11.7.4.	Desempeño	39
11.7.5.	Portabilidad.....	39
11.8.	Diagramas General de Casos de uso.....	40
11.9.	Arquitectura de desarrollo del software.....	45
11.9.1.	Modelo Cliente/Servidor	45
11.9.4.	Tipología del sistema.....	47
11.9.5.	Modelos de las Base de Datos	48

11.9.6.	Pruebas del sistema.....	50
11.9.7.	Checklist de Aceptación de Pruebas.....	53
11.9.8.	Ventanas principales del sistema.....	55
11.9.9.	Herramientas de desarrollo de la propuesta.....	58
11.10.	Motor de base de datos:	58
11.10.1.	SQL Server 2008	58
12.	IMPACTOS TÉCNICOS, SOCIALES, AMBIENTALES O ECONÓMICOS.	59
13.	PRESUPUESTO PARA EL DESARROLLO DE LA INVESTIGACIÓN.....	60
14.	CONCLUSIONES Y RECOMENDACIONES	61
15.	BIBLIOGRAFÍA	62
16.	ANEXOS	64

ÍNDICE DE TABLAS

Tabla 1. Beneficiarios.....	4
Tabla 2. Actividades y sistemas de tareas en relación a los objetivos.....	7
Tabla 3. Técnicas e instrumentos	32
Tabla 4. Diseño experimental.....	32
Tabla 5. Requerimientos del sistema.....	34
Tabla 6: Validación de Usuario	34
Tabla 7. Consulta de información	35
Tabla 8. Modificar información.....	35
Tabla 9. Consultas	35
Tabla 10. Reportes	36
Tabla 11. Interfaz.....	36
Tabla 12. Desempeño	36
Tabla 13. Seguridad en la Información	37
Tabla 14. Presupuesto.....	60

ÍNDICE DE FIGURAS

Figura 1.	Diagrama asignar equipo.....	40
Figura 2.	Diagrama selección equipo.....	41
Figura 3.	Diagrama ingresar usuario.....	42
Figura 4.	Diagrama Uso de Login.....	43
Figura 5.	Diagrama Inventario Equipo.....	44
Figura 6.	Modelo Lógico.....	48
Figura 7.	Modelo físico.....	49
Figura 8.	Prueba de Caja negra.....	52
Figura 9.	Prueba de Caja blanca.....	53
Figura 10.	Inventario de equipos de cómputo.....	55
Figura 11.	Equipos de cómputo.....	55
Figura 12.	Datos generales del equipo.....	56
Figura 13.	Cambio de Periféricos.....	56
Figura 14.	Cambio de Hardware.....	57
Figura 15.	Mantenimiento.....	57
Figura 16.	Historial del Equipo.....	58
Figura 17.	Forma de registro de información.....	82
Figura 18.	Registro de equipos.....	84
Figura 19.	Préstamo de equipos.....	84
Figura 20.	Información de equipos.....	85
Figura 21.	Sistema de registros de equipo.....	85
Figura 22.	Sistema de gestión de equipos.....	86
Figura 23.	Organización de la información.....	86
Figura 24.	Sistema de información.....	87
Figura 25.	Inicio de sesión.....	89
Figura 26.	Selección de lugar.....	89
Figura 27.	Cambio de contraseña.....	90
Figura 28.	Tipo de componente.....	90
Figura 29.	Selección del lugar.....	91

Figura 30	Procedimiento para eliminar registro.....	91
Figura 31	Inicio de nuevo ingreso de equipos.....	92
Figura 32	Ingreso de nuevos datos.....	92
Figura 33	Elección de bloque académico.....	93
Figura 34	Cambio de periférico.....	93
Figura 35	Datos de cambio de hardware.....	94
Figura 36	Datos del mantenimiento.....	94
Figura 37	Historial de equipos.....	95
Figura 38	Información de datos.....	95
Figura 39	Datos del mantenimiento.....	96

1. INFORMACIÓN GENERAL

Título del proyecto

“DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ”

Fecha de inicio: 10 de Octubre del 2016

Fecha de finalización: Agosto del 2017

Lugar de ejecución: Universidad Técnica de Cotopaxi Extensión La Maná.

Facultad que auspicia: Facultad de Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia: Ingeniería en Informática y Sistemas Computacionales

Proyecto de investigación vinculado: Universidad Técnica de Cotopaxi Extensión La Maná

Equipo de trabajo:

Coordinador de Proyecto: Rufino Yimael Chamaidan Panchana

Correo: rufino.chamaidan3@utc.edu.ec

Teléfono: 0982898601

Tutor: Ing. Henry Chanatasig MSc.

Correo: henry.chanatasig@utc.edu.ec

Área de conocimiento: Desarrollo de software

Línea de investigación: Tecnologías de la Información y comunicación (TICS) y Diseño Gráfico

Sub líneas de investigación de la carrera: Ingeniería de software

2. RESUMEN DEL PROYECTO

Estar conscientes de los avances tecnológicos hace notar que en los últimos años han contribuido sobremedida al desarrollo de la sociedad, es así que gran parte de las instituciones públicas y privadas se empeñan en aplicar nuevas tecnologías que aporten en la mejora sustancial de sus procesos, sin duda alguna la Universidad Técnica de Cotopaxi Extensión La Maná busca mejorar y sistematizar los procesos con el propósito de alcanzar un mayor nivel de eficiencia, de tal manera se propuso desarrollar un sistema informático para la gestión de equipos tecnológicos de la institución que permita mejorar la forma de registro del historial de mantenimiento tanto preventivo como correctivo de los equipos, además que brinde información de la dependencia de ubicación, responsable, componentes internos como externos, la marca y el software de los equipos; el sistema se desarrolló con herramientas específicas para el caso como son: case para el modelado de la base de datos, un motor de base de datos SQLServer, la programación se efectuó con el lenguaje de programación visual.net para el desarrollo de la interfaz gráfica; el sistema proporcionará información completa sobre todos los equipos tecnológicos con que cuenta la Universidad Técnica de Cotopaxi Extensión La Maná, el aporte que esta aplicación presta es la facilidad de obtener reportes sobre toda la información relacionada a los bienes informáticos que están a cargo del departamento de servicios informáticos, otras de las funcionalidades que brinda es la determinación de su vida útil y por ende dar de baja a un equipo que ya cumplieron con su vida útil, la aplicación se desarrollará con la metodología de desarrollo de software Scrum debido a que presta todas las guías para el desarrollo de los sistemas.

Palabras claves: *Sistema informático, Herramientas Case, SqlServer, Visual.net, Interfaz Grafica*

3. JUSTIFICACIÓN DEL PROYECTO

La presente investigación tuvo como finalidad satisfacer una de las necesidades prioritarias que presentó la Universidad Técnica de Cotopaxi, Extensión La Maná: establecer un sistema automatizado de gestión de control sobre los equipos tecnológicos, actualmente dicho proceso de es realizado de forma manual, en consecuencia se suscitan inconvenientes por la pérdida de información, ya que no existen respaldos, aquello imposibilita acceder a información fehaciente y ágil, puesto que se deben acudir al archivo físico y este proceso ocasiona demora para el solicitante de información como el responsable de otorgarla.

El sistema informático de gestión de equipos tecnológicos representó un gran aporte puesto que brindó apoyo a cada uno de los procesos que se desarrollan en el departamento informático, permite registrar, ordenar y clasificar la información que a diario se genera en relación al historial de mantenimiento preventivo y correctivo de los equipos, además brinda información de la dependencia de ubicación, responsables, componentes internos como externos, marcas, software de los equipos, vida útil del equipo, entre otras características.

La propuesta surgió como alternativa de solución frente a la problemática observada en la Universidad Técnica de Cotopaxi Extensión La Maná, se planteó desarrollar la presenta propuesta, a fin de solucionar los problemas detectados. Con la implementación de este aplicativo, mejoró el proceso de almacenamiento del historial de cada equipo tecnológico que actualmente se encuentra recopilado de forma manual, transformándolo a un proceso sistematizado, ya que hoy en día gracias a los avances tecnológicos existe la posibilidad de administrar la información en forma digital y así elevar el nivel de eficacia.

4. BENEFICIARIOS DEL PROYECTO

Satisfacer un conjunto de necesidades es la gran tarea de las nuevas tecnologías y que estas sean una solución a largo plazo y beneficien a un sector determinado de la sociedad es el elemento fundamental de las mismas, es esencial señalar que la presente investigación contribuirá con dos sectores de beneficiarios como se muestra en la siguiente tabla.

Tabla 1. Beneficiarios

BENEFICIADOS DIRECTOS	CANTIDAD
Servicio Informático	1
BENEFICIADOS INDIRECTOS	
Docentes	51
Empleados	15
TOTAL	67

Fuente: Secretaria U.T.C Extensión La Maná

Elaborado por: El autor

5. PROBLEMA DE INVESTIGACIÓN

Los avances tecnológicos que en los actuales momentos experimenta el mundo son de mucha importancia para el desarrollo de la sociedad, como uno de estos avance se puede mencionar el desarrollo de sistemas de gestión de aparatos tecnológicos, identificado en la Universidad Austral de Chile, el mismo que ha impulsado cambios permitiendo sistematizar los procesos manuales inadecuados y que ocasionan la perdida de información, debido a que en tiempos anteriores se efectuaba de forma manual.

En varias de las Instituciones de Educación Superior del país actualmente poseen sistemas informáticos que permiten gestionar de forma transparente los equipos tecnológicos, como ejemplo se puede señalar el de la Universidad de Guayaquil que maneja un sistema de gestión, en donde su principal función es la de llevar un control, registro y organización de todos los equipos tecnológicos que forman parte de la institución, manteniendo un registro de los componentes del equipo, en qué lugar se encuentran, el periodo de mantenimiento preventivo como correctivo, y sobre todo obtener reportes físicos de los equipos.

En el caso de la Universidad Técnica de Cotopaxi Extensión La Maná se puede apreciar que no cuenta con ningún tipo de sistema que permita la gestión eficiente de los equipos tecnológicos con los que cuenta la institución, los inconvenientes que se han podido identificar son: que el administrador de los laboratorios no conoce con exactitud que equipos se encuentran en las diferentes dependencias, no se está al tanto bajo qué persona se encuentra la custodia de un determinado equipo, se desconoce que equipos fueron sujetos a los diferentes tipos de mantenimiento, de idéntica manera no se puede conocer los componentes que les ha cambiado a los diferentes equipos tecnológicos.

Ante la problemática anteriormente expuesta es necesaria la ejecución de la propuesta planteada debido a que la misma será la solución a los inconvenientes identificados, luego de la puesta en marcha del sistema de gestión de equipos tecnológicos los beneficios serán evidentes como por ejemplo: se podrá conocer con exactitud la localización de cada uno de los equipos se encuentran en las diferentes dependencias, se estará al tanto bajo qué persona se encuentra la custodia de un determinado equipo, se conocerá que equipos fueron sujetos a los diferentes tipos de mantenimiento, de igual manera el administrador tendrá en detalle los componentes que les ha cambiado a los distintos equipos.

6. OBJETIVOS

6.1. Objetivo General

- Desarrollar un Sistema Informático que favorezca la gestión de equipos tecnológicos de la Universidad Técnica de Cotopaxi Extensión La Maná.

6.2. Objetivos Específicos

- Analizar los requerimientos que permitan establecer las necesidades esenciales para el desarrollo del sistema.
- Indagar sobre las metodologías de desarrollo de software para su adecuada aplicación en el proceso de desarrollo.
- Identificar las herramientas adecuadas para la ejecución de la propuesta.
- Implementar el sistema en base a pruebas de caja negra y caja blanca para corregir posibles fallas en su funcionamiento.

7. SISTEMA DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS

Tabla 2. Actividades y sistemas de tareas en relación a los objetivos

OBJETIVO	Actividad (tareas)	Resultado de la actividad	Descripción de la actividad (técnicas e instrumentos)
Analizar los requerimientos que permitan establecer las necesidades esenciales para el desarrollo del sistema.	Sesión de trabajo con el Responsable del departamento	Análisis de requerimiento de la información	Se formula un cuestionario de preguntas
Indagar sobre las metodologías de desarrollo de software para su adecuada aplicación en el proceso de desarrollo.	Recopilación bibliográfica sobre la metodología Extreming	Selección de la metodología de desarrollo de software XP Extreming	Investigación de las metodologías ágiles de desarrollo de software
Identificar las herramientas adecuadas para la ejecución de la propuesta.	Recopilación bibliográfica	Selección de las herramientas de desarrollo de software adecuada	Conjunto de herramientas de desarrollo de software adecuadas.
Implementar el sistema en base a pruebas de caja negra y caja blanca para corregir posibles fallas en su funcionamiento.	Herramientas de caso de caso de uso	Sistema de gestión instalado y expuesto a prueba.	Sistema de gestión validado y en funcionamiento

Elaborado por: El autor

8. FUNDAMENTACIÓN CIENTÍFICO TÉCNICA

8.1. Software y su ciclo de vida

Sin duda una de las tareas más complejas que puede abordar un profesional de la informática es el desarrollo de un sistema software, debido a la multitud de tareas que hay que llevar a cabo, desde el estudio previo de los requisitos, hasta su definitiva implementación y puesta en marcha. La ingeniería del software es el marco de referencia para el estudio de las diferentes alternativas con las que contamos para el desarrollo de productos de calidad estudiando las herramientas las metodologías y procedimientos empleados en proyectos de desarrollo de sistemas software (Fernández, 2012).

La norma IEEE 1074 define el ciclo de vida software como una aproximación lógica a la adquisición. El suministro, el desarrollo, la explotación y el mantenimiento software tras que la norma ISO 12207-1 entiende por modelo de ciclo de vida un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de producto de software, abarcando la vida del sistema desde la definición de los hasta la finalización de su uso (Rivera, 2012).

La norma ISO 12207-1 describe actividades que se pueden realizar durante el ciclo de vida del software y las agrupa en cinco procesos principales, ocho procesos de soporte y cuatro procesos generales (de la organización), así como un proceso que permite adaptar el ciclo de vida a cada concreto.

8.1.1. Modelos de ciclo de vida del software

8.1.1.1. El modelo en cascada

Es el ciclo de vida clásico y se deriva de otras ingenierías, se denomina así porque para comenzar una fase del ciclo debemos completar la fase anterior. Las fases en que se divide este ciclo de vida pueden presentar ligeras variaciones:

8.1.1.2. Análisis del sistema

Un sistema software forma parte de un sistema mayor con el cual se relaciona, en este punto analizamos el sistema en su conjunto para poder establecer objetivos y poder determinar cuáles de ellos deben abordables por el software (Leyva y Prieto, 2012).

Análisis de requisitos software.- En esta fase nos centramos en los requisitos del software definiendo las funciones a realizar, los datos el comportamiento y la interacción entre los elementos funcionales (Zapata, 2013).

Diseño.- En esta otra tomamos requisitos definidos en la etapa anterior y diseñamos los componentes del sistema de software a construir indicando las estructuras de datos a emplear, módulos a construir, procedimientos, algoritmos e interfaces que forman el sistema final (Zapata, 2013).

Codificación.- Tomando la documentación generada en el paso anterior debemos implementar el sistema en una máquina concreta codificando los procedimientos definidos en un lenguaje de programación, construyendo las interfaces y las estructuras de datos necesarias (Rivera, 2012).

Prueba.- Tras la codificación someter a prueba el sistema para cerciorarnos de su buen funcionamiento y del cumplimiento de los requisitos fijados al inicio del proyecto (Ramos, Noriega y Laínez, 2017).

Mantenimiento.- Producción de software no finaliza con la entrega del mismo ya que debe ser revisado para ajustarlo a nuevos requisitos, corrección de errores delectados tras la entrega o por la aparición de requisitos no contemplados en el proyecto inicial, y que hace necesaria la revisión del producto (Cisneros, 2015).

8.2. Metodología de desarrollo Scrum

8.2.1. ¿Qué es Scrum?

Es como un salvavidas para aquellas empresas que enfrentan dificultades al seguir la metodología de Cascada o que ni siquiera están usando metodología alguna para desarrollar su software. Scrum es un marco de referencia para crear software complejo y entregarlo a tiempo de una forma mucho más sencilla (Rivera, 2012).

Scrum es realmente fácil de entender, pero puede tomar algunos años antes de dominarlo. Sin embargo, esto no debe ser motivo para desinteresarse; los beneficios superan con creces la curva de aprendizaje. Scrum viene siendo usado desde principios de 1990 y actualmente ha ganado bastante popularidad. Como ya lo he mencionado antes. Scrum no es una metodología; es un marco de referencia dentro de la metodología de desarrollo de software Agile, el cual lo habilitará para crear excelente software, mediante la aplicación de un conjunto de directrices a seguir por los equipos de trabajo y el uso de roles concretos (Geribeth, 2010).

El marco de referencia Scrum utiliza el concepto de Equipos Scrum, los cuales son grupos de trabajo donde los miembros juegan roles específicos. Scrum considera que los desarrolladores de software son seres humanos que cometen errores, que piensan en nuevas ideas en el camino, y muchas características más.

8.3. Proceso de desarrollo de software

La ingeniería de software es una forma de ingeniería que aplica los principios de la ciencia de la computación y de la matemática para alcanzar con una mejor relación entre el coste y el beneficio para el problema de software, del mismo modo se trata de la aplicación sistemática, disciplinada y cuantificable para el desarrollo, y mantenimiento de software (Cisneros, 2015).

Tras un periodo de bonanza la crisis del software se identificó en los años sesenta sin embargo aún a día de hoy se notan sus efectos. Básicamente la crisis del software se fundamenta en los

problemas para entregar programas sin defectos o errores fáciles de entender y que sean verificables (Bueno, 2011).

Un proceso de desarrollo de software es una estructura utilizada para el desarrollo de un producto de software entre sus sinónimos están ciclo de vida y "proceso de software, hay muchos modelos para estos procesos, cada uno de ellos describiendo enfoques diferentes para una variedad de tareas y actividades a ejecutadas a lo largo del proceso.

8.3.1. Pasos del proceso de desarrollo de software

A continuación se muestran los pasos a ejecutar durante el desarrollo de software:

- Investigar los requisitos de los usuarios.
- La gran parte de usuarios por no decir todos no saben exactamente lo que ellos quieren. Esto se debe a que la mayoría no sabe cuáles son exactamente las acciones que se llevan a cabo a lo largo del día.
- El análisis requiere que el desarrollador se convierta intencionalmente en un especialista en el dominio del usuario para ayudarlo y guiarlo en la definición de sus requisitos (Cisneros, 2015).

Podemos dividir esta fase en cuatro apartados: observación y descubrimiento de la máxima información, interrogar y tratar de aclarar al máximo la información seguidamente el desarrollador comprobar la información y soluciones; redactar del problema y realizar un documento con la especificación de requisitos (Cisneros, 2015).

- Definir claramente las características necesarias para el sistema (especificación)

La especificación de requisitos es la última fase de la tarea del análisis. Necesita recoger de forma no ambigua cual es el comportamiento requerido. En el documento se recogen notaciones formales, documentos estructurados y ejemplos (Geribeth, 2010).

- Crear o adaptar una solución adecuada al problema, es decir, la creación del proyecto.
- El proyecto busca desarrollar una solución que atienda a los requisitos, con base en la experiencia acumulada (y técnicas estandarizadas).

- El resultado final es un de proyecto que de forma ambigua comunica el proyecto a aquellos que lo irán a implementar.
- Desarrollar la solución propuesta (implementación)

Durante esta fase se lleva a cabo el de la aplicación en si misma. Es el momento de escribir el código, documentarlo, solucionar cualquier error que se detecte, preparar el código para ser testado, enviar informaciones tanto al proyectista como al analista, así como al testador y/o integrador. El objetivo es alcanzar el código de trabajo y la documentación asociada actualizado listo para ser probado (Leyva y Prieto, 2012).

8.3.2. Categorías de paradigmas de programación

A continuación se muestran los principales paradigmas de programación:

- Los que soportan técnicas de programación de bajo nivel (ejemplo: copia de ficheros frente estructuras de datos compartidos)
- Los que soportan métodos de diseño de algoritmos (ejemplo: divide y vencerás, programación dinámica, etc.)
- Los que soportan soluciones de programación de alto nivel, como los descritos en el punto anterior.
- Los paradigmas relacionados con la programación de alto nivel se agrupan en tres categorías de acuerdo con la solución que aportan para resolver el problema:
- Solución procedimental u operacional. Describe etapa a etapa el modo de construir la solución. Es decir señala la forma de obtener la solución.
- Solución demostrativa, es una variante de la procedimental especifica la solución describiendo ejemplos y permitiendo que el sistema generalice la solución de estos ejemplos para otros casos.
- Aunque es fundamentalmente procedimental, el hecho de producir resultados muy diferentes a ésta, hace que sea tratada como una categoría separada.

- Solución declarativa. Señala las características que debe tener la solución, sin describir cómo procesarla. Es decir señala qué se desea obtener pero no cómo obtenerlo (Ramos, et. al., 2017).

8.3.3. Diagramas de flujos de datos

El DFD (Data Flow Diagram) surgió de la necesidad de un nuevo método de especificación sencillo de implantar, fácil comprensión y comunicación. El DFD fue desarrollado en los años 70's y fue popularizado por Yourdan. Es un método de especificación utilizado hasta la fecha. Para empezar se puede preguntar ¿Que son los diagramas de flujos de Datos?. Un diagrama de flujo de datos (DFD) es una representación gráfica de los procesos que se realizan con los datos en su organización, con el uso de tan solo cuatro símbolos, se puede crear una descripción grafica de los procesos que, con el tiempo, contribuirán a desarrollar una sólida documentación del sistema (Terán, 2014).

- En seguida mencionan las ventajas sobre las explicaciones descriptivas en relación con la forma en que los datos se mueven a través del sistema:
- Libertad para emprender la implementación técnica del sistema en las primeras etapas.
- Comprensión más profunda de la interrelación entre sistemas y subsistemas.
- Comunicación con los usuarios sobre el conocimiento del sistema actual mediante diagramas de flujos de datos.
- Análisis de un sistema propuesto para determinar si se han definido los datos y procesos necesarios.
- Proporciona liberación en las etapas que se realizan (Fernández, 2012).

8.4. Sistema de control

Se definido como un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado, de modo que se reduzcan las probabilidades de fallos y se obtengan los resultados buscados (Bueno, 2011).

Un sistema está integrado por una serie de elementos que actúan conjuntamente y que cumplen un cierto objetivo. Los elementos que componen un sistema no son independientes, sino que

están estrechamente relacionados entre sí, de forma que las modificaciones que se producen en uno de ellos pueden influir en los demás (Geribeth, 2010).

Una vez analizado los sistemas de control se entiende que están desarrollados para manejar procesos, de modo que se pueda llevar un adecuado registro de la información que se maneja en una organización o empresa, reduciendo así posibles fallos y logrando resultados favorables.

8.5. SQL server 2008

Una base de datos en un sistema relacional está compuesta por un conjunto de tablas, que corresponden a las relaciones del modelo relacional. En la terminología usada en SQL no se alude a las relaciones, del mismo modo que no se usa el término atributo, pero sí la palabra columna. A continuación, se usarán indistintamente ambas terminologías, por lo que tabla estará en lugar de relación, columna en el de atributo y línea en el de tabla, y viceversa, la creación de la base de datos consiste en la creación de las tablas que la componen (Fernández, 2012).

En realidad, antes de poder proceder a la creación de las tablas, normalmente hay que crear la base de datos, lo que a menudo significa definir un espacio de nombres separado para cada conjunto de tablas. De esta manera, para una DBMS se pueden gestionar diferentes bases de datos independientes al mismo tiempo sin que se den conflictos con los nombres que se usan en cada una de ellas (Bueno, 2011).

Management Studio está diseñado para desarrollar y administrar objetos de base de datos y para administrar y configurar los objetos existentes de AnalysisServices. Si está implementando una solución que utiliza servicios de bases de datos de SQL Server o si está administrando una solución existente que utiliza SQL Server, AnalysisServices, IntegrationServices o ReportingServices, debe utilizar Management Studio. El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones en éstos últimos.

8.5.1. Características en SQL Server 2008

La versión 2008 de SQL Server contiene numerosas novedades tanto para el administrador de la base de datos como para el desarrollador de aplicaciones. Ofrece soporte de transacciones, procedimientos almacenados, incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente (Leyva y Prieto, 2012).

Permite trabajar y administrar información de otros servidores de datos 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 información.

8.5.2. Ventajas SQL Server 2008

Los privilegios a nivel de la base de datos posibilitan extraer y actualizar datos, crear objetos como procedimientos y tablas:

- SQL server solo tiene un sistema de almacenamiento para todo, sin embargo es más seguro que MySQL durante un corte de energía, ya que hace varios controles para mantener la integridad de los datos.
- SQL SERVER tiene varias herramientas de reporte y análisis.
- SQL SERVER es más fácil de instalar y de manejar. Y viene en diferentes versiones.
- Menor necesidad de la limpieza de las memorias internas (Terán, 2014).

8.5.3. Desventajas SQL Server 2008

- SQL Tiene soporte solo con el sistema operativo Windows
- Tiene límite de conexiones simultáneas.
- Utiliza una enorme cantidad de RAM.
- Costo de la licencia (Leyva y Prieto, 2012).

8.5.4. Servidor de Base de datos SQL Server 2008

Es un sistema de gestión de base de datos producido por Microsoft basado en el modelo relacional (SGBD) asentado en el lenguaje Transact-SQL, específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea (Geribeth, 2010).

El motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos, proporciona acceso controlado y procesamiento rápido de transacciones para cumplir los requisitos de las aplicaciones consumidoras de datos más exigentes de su empresa (Zabala, 2011).

El Motor de base de datos también proporciona una completa compatibilidad para mantener una gran disponibilidad, se puede decir que el motor de base de SQL server es lo más recomendable para realizar sistema de base de datos, ya que nos proporciona grandes capacidades para almacenar información, así mismo es el más utilizado para trabajar en sistemas clientes servidores con un rendimiento fiable y oportuno, permitiendo tener acceso controlado a los datos con el único propósito de mantener la seguridad y la estabilidad.

8.6. Visual Studio 2010

Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes (Fernández, 2012).

Tales lenguajes utilizan las funciones de NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML; facilita un conjunto de tecnologías y herramientas que simplifican el desarrollo de aplicaciones y propone una solución para cada tipo de aplicaciones, la independencia de lenguaje y la transparencia a través de redes.

Aplicaciones Windows clásicas

Aplicaciones web

Servicios Windows

Servicios web (Zapata, 2013).

Todos estos tipos de aplicaciones se los puede realizar gracias a un elemento esencial que es el Framework, se encarga mediante numerosas capas de software superpuesto, de la integridad de la vida de una aplicación, desde el desarrollo hasta la ejecución. Debe estar instalado en el sistema operativo con el cual va a interactuar, el primer sistema que permite acoger es el Windows, Linux y Unix. Visual Studio .NET es la herramienta de desarrollo multilenguaje más completa para construir e integrar rápidamente aplicaciones y servicios web XML. Aumenta de un modo extraordinario la productividad de los desarrolladores y crea nuevas oportunidades de negocio.

8.7. Metodología XP (Extreme Programming).

Surge ante la necesidad de ofrecer una alternativa a las metodologías tradicionales, caracterizadas por ser rígidos y dirigidos por la documentación que se genera. En cada una de las actividades desarrolladas (Geribeth, 2010).

Nace de la mano de Kent Beck en el verano de 1996, cuando trabajaba para Chrysler Corporation, las ideas primordiales de su sistema las comunicó en la revista C++ Magazine en una entrevista que ésta le hizo el año (Geribeth, 2010).

8.7.1. Descripción de Metodología (XP)

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado, está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software (Fernández, 2012).

Su objetivo es aumentar la productividad al desarrollar 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. El desarrollo de software usando XP, es fortalecido con cuatro valores importantes que son:

- Comunicación
- Simplicidad
- Valor
- Retroalimentación (Bueno, 2011).

8.7.2. Características de XP

- Es una metodología que se basa en la prueba y el error.
- Se fundamenta en valores y prácticas.
- Expresada en forma de 12 practicas-conjunto, que se soportan la una con la otra y trabajan todas en conjunto (Fernández, 2012).

8.7.3. Ventajas

- La programación es organizada.
- Los errores se reducen significativamente.
- Satisfacción del programador

Dentro de estas metodologías ágiles está la metodología XP (Extreme Programming), que es la metodología utilizada en este proyecto. Esta metodología permite desarrollar la aplicación en base a pruebas y errores, además de permitir una programación ordenada y en base a las expectativas del cliente. (Fernández, 2012).

8.8. Sistema cliente servidor

El modelo cliente-servidor (client-server), describe el proceso de interacción entre la computadora local (el cliente) y la remota (el servidor). El cliente le hace peticiones (solicitudes, requerimientos) al servidor, el cual procesa dicho requerimiento y retorna los resultados al cliente apropiado. Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma (Zapata, 2013).

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras (Rivera, 2012).

Por lo general, los sistemas clientes servidores son diseñados y creados para trabajar dentro de una red área local mediante de computadoras que se encuentre conectadas entre sí donde un equipo hace el papel de servidor y comparte información, intercambio de emails, el acceso a webs, el acceso a bases de datos, impresoras y muchos otros protocolos de internet, para que los clientes puedan trabajar desde su respectiva área y almacenar información en un solo servidor.

8.8.1. Características del modelo cliente/servidor

La máquina que sirve como servidor (host) es la que tiene en ejecución programas de servidor que contestan los requerimientos de los clientes. Por lo general los clientes inician la sesión de comunicación, mientras que los servidores esperan la llegada de solicitudes. La interacción entre cliente y servidor es generalmente representada empleando diagramas de secuencia estandarizados en UML (Rivera, 2012).

8.8.2. Ventajas del modelo cliente/servidor

El esquema cliente/servidor posee las siguientes ventajas:

- Uno de los aspectos que más ha promovido el uso de sistemas cliente/servidor, es la existencia de plataformas de hardware cada vez más baratas, que las requeridas por una solución centralizada, basada en sistemas grandes.
- El esquema cliente/servidor facilita la integración entre sistemas diferentes y comparte información permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario.
- Una ventaja adicional del uso del esquema cliente/servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL).
- El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global (Leyva y Prieto, 2012).

8.8.3. Funciones del cliente

Son las máquinas o procesos que piden información, recursos y servicios a un servidor unido. Estas peticiones pueden ser cosas como proporcionar datos de una base de datos, aplicaciones, partes de archivos o archivos completos a la máquina cliente. (Zabala, 2011)

8.9. Herramientas para el diseño del sistema de control

8.9. 1. La replicación

Es un conjunto de tecnologías destinadas a la copia de distribución de datos y objetos de base de datos de una base de datos a otra, para luego sincronizar ambas bases de datos con el fin de mantener su coherencia. La réplica permite distribuir datos a diferentes ubicaciones y a usuarios remotos o móviles mediante redes de área local y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas en internet (Bueno, 2011)

8.9. 2. Herramientas CASE

8.9.2.1. Visual Paradigm

Es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Geribeth, 2010).

8.9.2.1.1. Propósito de Visual Paradigm

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software, la versión Visual Paradigm UML 6.4 Community Edition (Community Edition, ya que existe la Enterprise, Professional, etc). Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos (Leyva y Prieto, 2012).

8.9. 2. 1. 2. Características de Visual Paradigm

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- Disponibilidad de múltiples versiones, con diferentes especificaciones.
- Licencia: gratuita y comercial y soporta aplicaciones Web.
- Las imágenes y reportes generados, no son de muy buena calidad.

- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar y compatibilidad entre ediciones.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversión (control de versiones).
 - ✓ Interoperabilidad con modelos UML2
 - ✓ Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML, NET exe/dll, CORBA IDL.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Generación de código y despliegue de EJB - Generación de beans para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos.
- Soporte ORM - Generación de objetos Java desde bases de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Editor de figuras (Cisneros, 2015).

8.9.3. Lenguaje de programación

C#: Es un lenguaje de programación orientado a objetos y diseñado para la infraestructura de lenguaje común. Contiene 2 categorías generales de tipos de datos integrados: tipos de valor y

tipos de referencia. El término tipo de valor indica que esos tipos contienen directamente sus valores (Zapata, 2013).

8.9.3. 1. Sencillez de uso

Dicha se le confiere porque C# elimina muchos elementos añadidos por otros lenguajes y que facilitan su uso y comprensión. Es por ello que se dice que C# es auto contenido (Zapata, 2013).

8.9.3. 2. Compatible

Para facilitar la migración de programadores de C++ o Java a C#, no sólo se mantiene una sintaxis muy similar a la de los dos anteriores lenguajes, sino que el CLR también ofrece la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos, tales como las DLLs de la API de Win32 (Zapata, 2013).

8.9.3. 3. Modernidad

C# un lenguaje de última generación, implementa elementos que se ha demostrado a lo largo del tiempo que son muy útiles para el programador, como tipos decimales o booleanos, un tipo básico string así como una instrucción que permita recorrer colecciones con facilidad. Estos elementos hay que simularlos en otros lenguajes como C++ o Java.

8.9.3. 4. Orientado a objetos

C# es orientado a objetos. C# no permite la inclusión de funciones ni variables globales que no estén incluidos en una definición de tipos, por lo que la orientación a objetos es más pura y clara que en otros lenguajes. C# soporta todas las características del paradigma de la programación orientada a objetos, como son la encapsulación, la herencia y el polimorfismo (Fernández, 2012).

8.9.3. 5. Recolección de basura

Como ya se comentó, todo lenguaje incluido en la plataforma .NET tiene a su disposición el recolector de basura del CLR. Esto implica que no es necesario incluir instrucciones de destrucción de objetos en el lenguaje (Fernández, 2012).

8.9.3.6. Extensión de los operadores básicos

Para hacer mucho más fácil la legibilidad de código y poder que los nuevos tipos de datos que se definan a través de las estructuras estén al mismo nivel que los elementos predefinidos en el lenguaje, a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores cuando se apliquen a diferentes tipos de objetos (Fernández, 2012).

8.9.3. 7. Seguridad de tipos

C# incluye formas de control de acceso a tipos de datos, lo que garantiza que no se produzcan errores difíciles de detectar como un acceso a memoria de ningún objeto. Para ello, el lenguaje provee de una serie de normas de sintaxis, como no realizar conversiones entre tipos que no sean compatibles, no se pueden usar variables no inicializadas previamente, y en el acceso atablas se hace una comprobación de rangos para que no se excedan ninguno de los índices de la misma (Zapata, 2013).

8.9.3.8. Instrucciones seguras

Para evitar errores comunes como se producían en otros lenguajes, se impusieron una serie de restricciones en el uso de instrucciones de control más comunes (Zapata, 2013).

8.9.3.9. Unificación de tipos

En C# todos los tipos derivan de una superclase común llamada System Object, por lo que automáticamente heredarán todos los miembros definidos en esta clase. Es decir, son objetos. A diferencia de Java, en C# esta característica también se aplica para los tipos básicos (Leyva y Prieto, 2012).

8.9.3.10. Extensión de los operadores básicos

Para hacer mucho más fácil la legibilidad de código y poder que los nuevos tipos de datos que se definan a través de las estructuras estén al mismo nivel que los elementos predefinidos en el lenguaje, a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores (incluidos el de la conversión) cuando se apliquen a diferentes tipos de objetos (Rivera, 2012).

8.9.3. 11. Eficiente

En C#, todo el código incluye numerosas restricciones para garantizar su seguridad, no permitiendo el uso de punteros. Existen modificadores para saltarse esta restricción, pudiendo manipular objetos a través de punteros. Para ello basta identificar regiones de código con el identificador, y podrán usarse en ellas punteros de forma similar a como se hace en C++. (Rivera, 2012).

8.9.4. Metodología Scrum

Es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa (ROI). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación. Con la metodología Scrum el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema (Geribeth, 2010).

Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades (Fernández, 2012).

8.9.4. 1. Beneficios

Cumplimento de expectativas.- El cliente establece sus expectativas indicando el valor que le aporta cada requisito / historia del proyecto, el equipo los estima y con esta información el Product Owner establece su prioridad. De manera regular, en las demos de Sprint el Product Owner comprueba que efectivamente los requisitos se han cumplido y transmite se feedback al equipo (Leyva y Prieto, 2012).

Flexibilidad a cambios.- Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos (Leyva y Prieto, 2012, pág. 42).

Reducción del Time to Market.- El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo (Leyva y Prieto, 2012).

Mayor calidad del software.- La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.

Mayor productividad.- Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse (Leyva y Prieto, 2012).

Maximiza el retorno de la inversión (ROI).- Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión (Leyva y Prieto, 2012).

Predicciones de tiempos.- Mediante esta metodología se conoce la velocidad media del equipo por sprint (los llamados puntos historia), con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está en el Backlog (Leyva y Prieto, 2012).

Reducción de riesgos.- El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada (Leyva y Prieto, 2012).

8.9.4. 2. Proceso de Scrum

- **El proceso.-** El desarrollo se realiza de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas. En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizándose siempre aquellas que aporten mayor valor de negocio (Geribeth, 2010).
- **Product Backlog.-** Conjunto de requisitos denominados historias descritos en un lenguaje no técnico y priorizados por valor de negocio, o lo que es lo mismo, por retorno de inversión considerando su beneficio y coste. Los requisitos y prioridades se revisan y ajustan durante el curso del proyecto a intervalos regulares (Geribeth, 2010).
- **Sprint Planning.-** Reunión durante la cual el Product Owner presenta las historias del backlog por orden de prioridad. El equipo determina la cantidad de historias que puede comprometerse a completar en ese sprint, para en una segunda parte de la reunión, decidir y organizar cómo lo va a conseguir (Geribeth, 2010).
- **Sprint.-** Iteración de duración prefijada durante la cual el equipo trabaja para convertir las historias del Product Backlog a las que se ha comprometido, en una nueva versión del software totalmente operativo (Geribeth, 2010).
- **Sprint Backlog.-** Lista de las tareas necesarias para llevar a cabo las historias del sprint.
Daily sprint meeting: Reunión diaria de cómo máximo 15 min. En la que el equipo se sincroniza para trabajar de forma coordinada. Cada miembro comenta que hizo el día anterior, que hará hoy y si hay impedimentos (Bueno, 2011).

- **Demo y retrospectiva.-** Reunión que se celebra al final del sprint y en la que el equipo presenta las historias conseguidas mediante una demostración del producto. Posteriormente, en la retrospectiva, el equipo analiza qué se hizo bien, qué procesos serían mejorables y discute acerca de cómo perfeccionarlos (Bueno, 2011).

En Scrum, el equipo se focaliza en construir software de calidad. La gestión de un proyecto Scrum se centra en definir cuáles son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en vencer cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo (Bueno, 2011).

8.9. 5. Roles de SCRUM

El equipo Scrum está formado por los siguientes roles:

Scrum master.- Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el ROI (Terán Pérez, 2014).

Product owner (PO).- Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y él es responsable del ROI del proyecto (entregar un valor superior al dinero invertido). Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las reprioriza de forma regular (Terán Pérez, 2014).

Team.- Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint (Terán Pérez, 2014).

8.9.6. Herramientas de Programación

8.9.6.1. Visual Studio 2010

Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML (Ramos, el al., 2017).

- **Entorno de desarrollo integrado**

Visual Studio 2010 Professional proporciona compatibilidad con varios monitores, de modo que pueda organizar y administrar su trabajo como quiera. También puede dar rienda suelta a su creatividad utilizando los diseñadores visuales para mejorar las últimas plataformas, incluido Windows 7 (Ramos, el al., 2017).

Compatibilidad con la plataforma de desarrollo

Tanto si un proyecto es sencillo como complejo, puede usar Visual Studio 2010 Professional para plasmar su visión de la vida en una serie de plataformas que incluyen Windows, Windows Server, Web, Cloud, Office y SharePoint, entre otras (Ramos, el al., 2017).

- **Herramientas de prueba**

Visual Studio 2010 Professional incluye capacidades para pruebas de unidades dentro del entorno de desarrollo integrado que pueden generar todos los componentes de métodos necesarios para compilar pruebas de unidades, lo que ayuda a garantizar que cada unidad de código funcione correctamente (Ramos, el al., 2017).

9. HIPÓTESIS

¿La implementación de un sistema informático permitirá mejorar la gestión los equipos tecnológicos en la Universidad Técnica de Cotopaxi Extensión La Mana?

10. METODOLOGÍA DE LA INVESTIGACIÓN

10.1. Investigación de campo

Es aquella en la que el mismo objeto de estudio sirve como fuente de información para el investigador.

10.2. Investigación bibliográfica y documental

Es aquella que la cual tiene que investigarse todo lo referente a sistemas de control de sistemas informáticos, o sea, aquella información que existe en documentos y material.

10.3. MÉTODOS DE INVESTIGACIÓN

10.3.1. Método inductivo

Consiste en basarse en enunciados singulares, tales como descripciones de los resultados de observaciones o experiencias para plantear enunciados universales, tales como hipótesis o teorías. Ello es como decir que la naturaleza se comporta siempre igual cuando se dan las mismas circunstancias, lo cual es admitir que bajo las mismas condiciones experimentales se obtienen los mismos resultados base de la repetitividad de las experiencias (Cegarra, 2012).

10.3.2. Método deductivo

Es más propio de las ciencias formales (como la matemática y la lógica), consiste en ir de lo general a lo particular, mediante el uso de argumentos y/o de silogismos. En él se utilizan la lógica para llegar a conclusiones, a partir de determinadas premisas. (Zarzar, 2015, pág. 64).

10.4. TÉCNICAS DE INVESTIGACIÓN

10.4.1. Entrevista

Es la herramienta metodológica preferida por los investigadores cualitativos. Esta suele ser practicada por los investigadores de esta escuela en su modalidad abierta y no estructurada, entendida como una conversación controlada por el arte de saber formular preguntas y de saber escucharlas. Dentro del paradigma constructivista, la entrevista no es un arma neutral, sino que el entrevistador crea la situación social de la entrevista, en cuyo marco cobran sentido las respuestas obtenidas y dentro del cual deben ser interpretadas como episodios interaccionales. (Ruiz Olabuénaga, 2012).

10.4.2. Encuesta

Es una técnica de recogida de datos mediante la aplicación de un cuestionario a una muestra de individuos. A través de las encuestas se pueden conocer las opiniones, las actitudes y los comportamientos de los ciudadanos.

En una encuesta se realizan una serie de preguntas sobre uno o varios temas a una muestra de personas seleccionadas siguiendo una serie de reglas científicas que hacen que esa muestra sea, en su conjunto, representativa de la población general de la que procede. (CIS, 2015)

10.5. DISEÑO EXPERIMENTAL

Tabla 3. Técnicas e instrumentos

N°	TECNICAS	INSTRUMENTOS
1	Encuestas	Cuestionario
2	Entrevista	Test

Elaborado por: El autor

Tabla 4. Diseño experimental

Agente y/o Tecnologías	Funciones	Técnicas, espacios de trabajo y difusión	Población	Muestra	Cantidad Tot.
Estudiante	Desarrollador		1	1	1
Docente	Tutor		1	1	1
Población	Docente universitario		1	1	1
Profesionales	Servicios Informáticos		1	1	1
				Total	4

Elaborado por: El autor

11. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

11.1. Muestra

Es aquella representación proporcional de las características de toda una población que es objeto de estudio, para establecer el cálculo respectivo se aplican formulas las cuales ayudan a simplificar el número de personas en una población extensa, pero existen casos que la población no supera el rango establecido, pues en esta investigación la población es inferior a los 100 individuos, en tal razón no es posible aplicar dicha fórmula en consecuencia y se trabajó con el total de la población existente.

11.2. Análisis de resultados de las encuestas

En base a las encuestas aplicadas es evidente que no existe ningún tipo de sistema que permita registrar la información de los equipos informáticos con los que cuenta la Universidad Técnica de Cotopaxi en relación de tecnologías de la información, por otro lado los encuestados manifestaron estar de acuerdo con la aplicación de un sistema que permita mejorar la gestión de los equipos, en consecuencia resulta de vital contar con un sistema que permita establecer mejoras en el control de los equipos de cómputo (**ver anexo 4**).

11.3. Requisitos mínimos del sistema

Memoria mínima	1GB
Espacio en disco	50 GB
Procesador	Intel Core I3
Sistema Operativo	Windows 7 32 a 64 bits

11.4. Requerimientos del sistema

Herramientas case:	Virtual Paradim
Lenguaje de programación:	C#
Metodología:	Scrum
Motor de base de datos:	Ms SQL Server 2008
Herramientas de desarrollo:	Visual Studio 2010

11.5. Requerimientos para el desarrollo del sistema

Como etapa primordial es necesario recopilar los elementos fundamentales que fueron facilitados por el responsable del Departamento de Tecnologías de la Información, los requerimientos se compaginan con cada una de las necesidades a cubrir.

Tabla 5. Requerimientos del sistema

REQUERIMIENTOS DEL CLIENTE	APROBACIÓN DEL CLIENTE
Almacenamiento de información	X
Ingresar datos	X
Actualizar datos	X
Consultas	X
Reportes	X
Interfaz grafica	X

Elaborado por: El autor

11.5.1. Requerimientos Funcionales

Tabla 6: Validación de Usuario

Requerimiento	RF01
Nombre	Validación de Usuario
Características propias	Validación obligatoria para ingresar al sistema
Descripción	El sistema permitirá el acceso a la información
Prioridad	Alta

Elaborado por: El autor

Tabla 7. Consulta de información

Requerimiento	RF02
Nombre	Consulta de información
Características propias	Información general de los equipos de computo
Descripción	Proporciona información general de los equipos
Prioridad	Alta

Elaborado por: El autor

Tabla 8. Modificar información

Requerimiento	RF03
Nombre	Modificar
Características propias	Permite modificar datos almacenados
Descripción	Permite modificar datos almacenados de los equipos
Prioridad	Alta

Fuente: El autor

Tabla 9. Consultas

Requerimiento	RF04
Nombre	Realizar Consultas
Características propias	Permite realizar consultas
Descripción	Permite a los usuarios verificar información almacenada de los equipos
Prioridad	Alta

Elaborado por: El autor

Tabla 10. Reportes

Requerimiento	RF05
Nombre	Reportes
Características propias	Permite obtener reportes
Descripción	Permite a los usuarios obtener reportes de formato digital a físico
Prioridad	Alta

Elaborado por: El autor

11.4.2. Requerimientos no funcionales

Tabla 11. Interfaz

Requerimiento	RNF01
Nombre	Interfaz
Características propias	Interfaz amigable con los usuarios
Descripción	Interfaz de fácil manejo para con los usuarios
Prioridad	Alta

Elaborado por: El autor

Tabla 12. Desempeño

Requerimiento	RNF02
Nombre	Desempeño
Características propias	El sistema proporcionara a los usuarios un adecuado desempeño en el almacenamiento de datos.
Descripción	La información estará disponible cuando los usuarios deseen hacer uso de la misma
Prioridad	Alta

Elaborado por: El autor

Tabla 13. Seguridad en la Información

Requerimiento	RNF03
Nombre	Seguridad en la información
Características propias	El sistema proporcionara a los usuarios seguridad en cuanto a la información almacenada.
Descripción	La información estará disponible y segura cuando los usuarios deseen hacer uso de la misma
Prioridad	Alta

Elaborado por: El autor

11.6. Funcionalidades del sistema

- Gestionar la serie de la pc, el monitor, el teclado y el mouse
- Consultar donde pertenece el equipo
- Consultar nombre de la persona encargada del equipo
- Registrar los detalles del mantenimiento preventivo y correctivo
- Consultar el estado del equipo
- Consultar el número de equipos dados de baja
- Consultar el número de equipos por dar de baja
- Consultar la ubicación del equipo en mantenimiento
- Consultar préstamos del equipo
- Registrar el código del equipo
- Registrar la hora entrada
- Registrar la hora salida
- Gestionar los componentes del equipo
- Consultar el nombre de la persona que lleva el equipo
- Registrar el precio del equipo
- El sistema debe ser capaz de procesar 1000 transacciones por segundo.
- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
- El sistema tiene la capacidad de operar adecuadamente varios usuarios de manera simultánea.

- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.
- Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
- El nuevo sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.
- Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptados utilizando el algoritmo RSA.

11.7. Requisitos de Rendimiento

- Garantizar que el uso del sistema no interfiera en el normal proceso de desempeño de la base de datos.
- El aplicativo debe tener la capacidad de manejar un gran volumen de información.

11.7.1. Usabilidad

El sistema debe poseer una interfaz amigable con el usuario

11.7.2. Seguridad

El sistema deberá contar con sus respectivas credenciales de acceso usuario y contraseña.

11.7.3. Multiplataforma

Su funcionamiento lo debe hacer en diferentes sistemas operativos

11.7.4. Desempeño

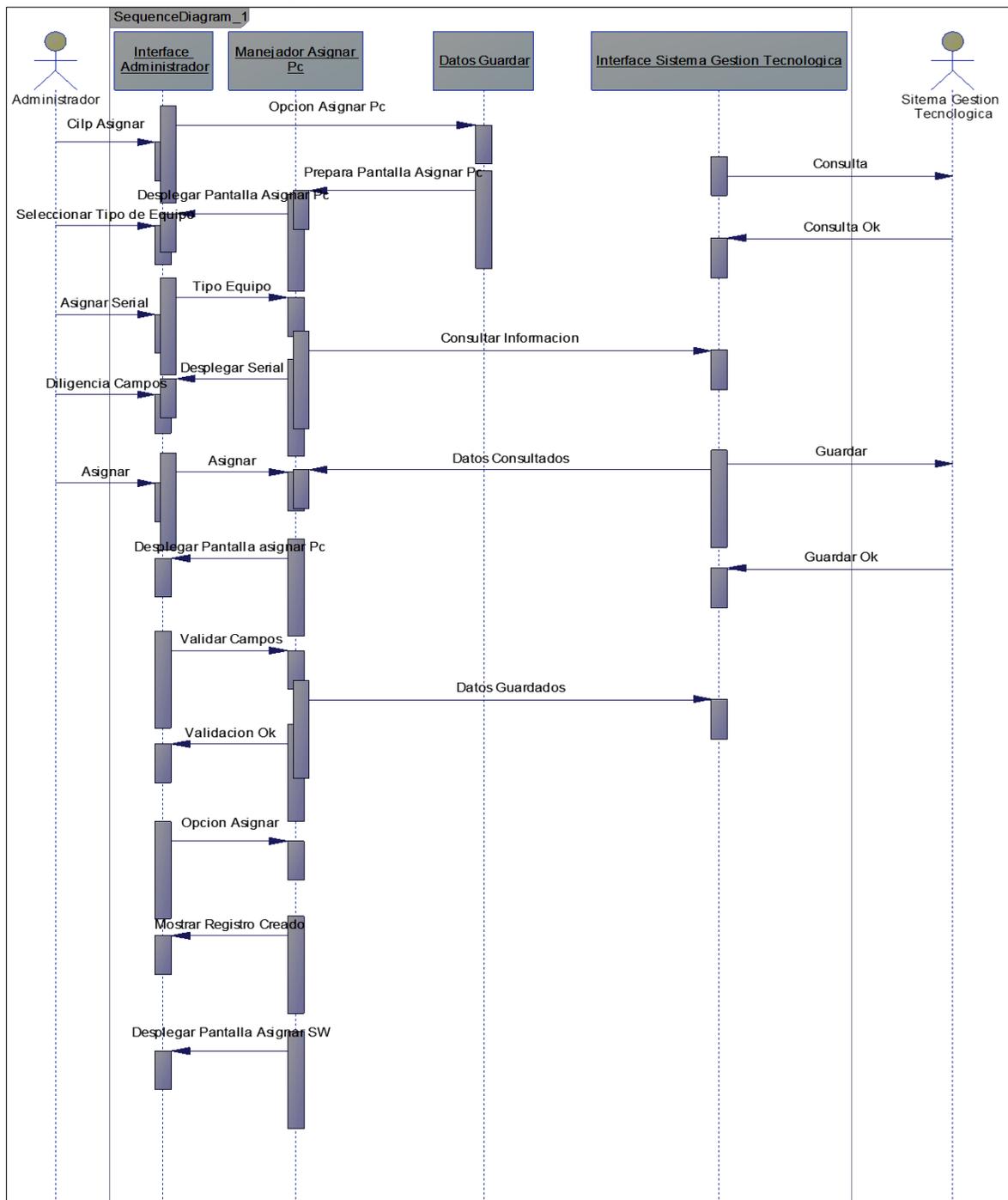
En el sistema no se presentarán problemas tanto en el manejo como en la implementación

11.7.5. Portabilidad

El aplicativo será implementado bajo el sistema operativo Windows

11.8. Diagramas General de Casos de uso

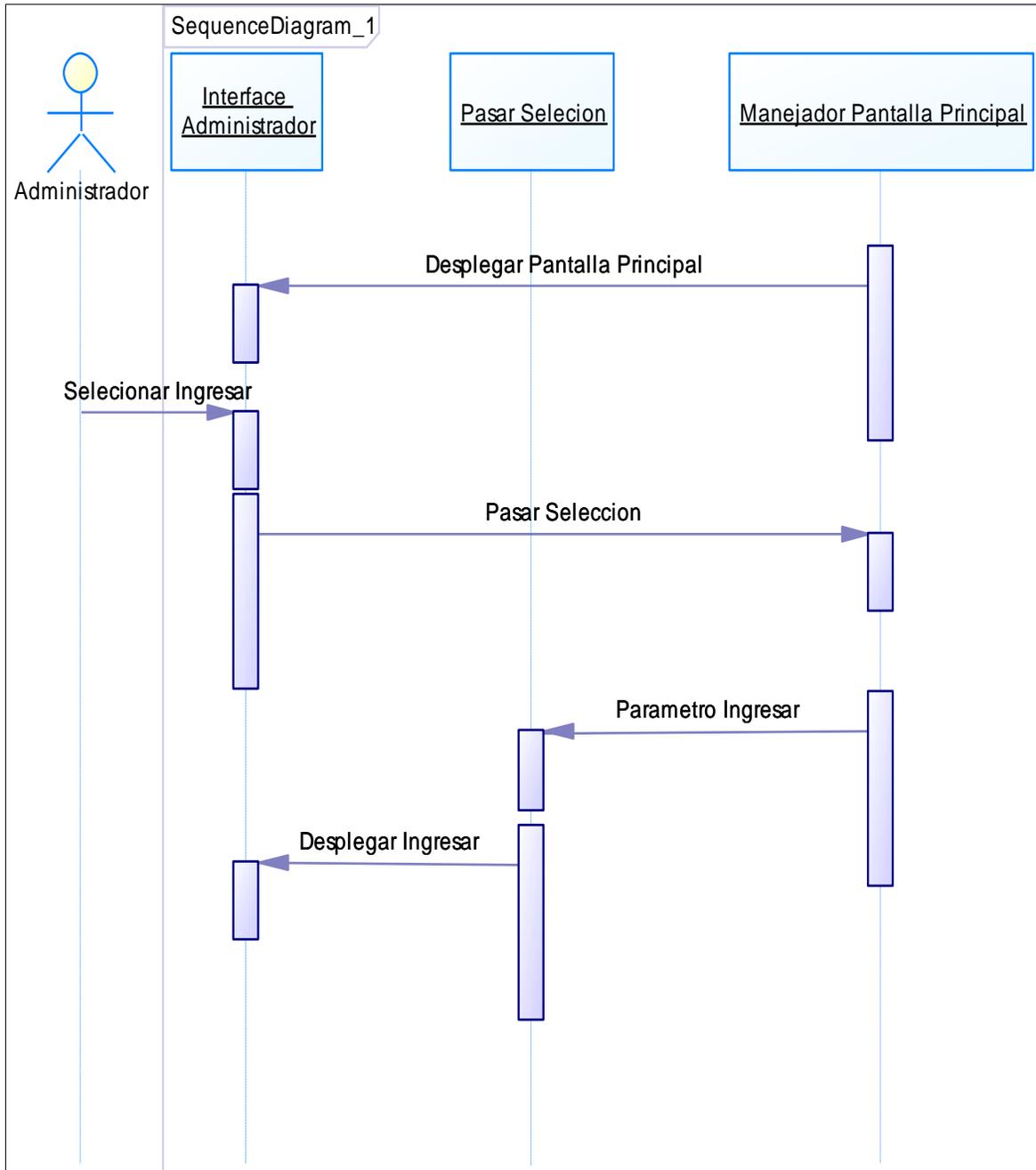
Figura 1. Diagrama asignar equipo



Elaborado por: El autor

DIAGRAMA PASAR SELECCIÓN

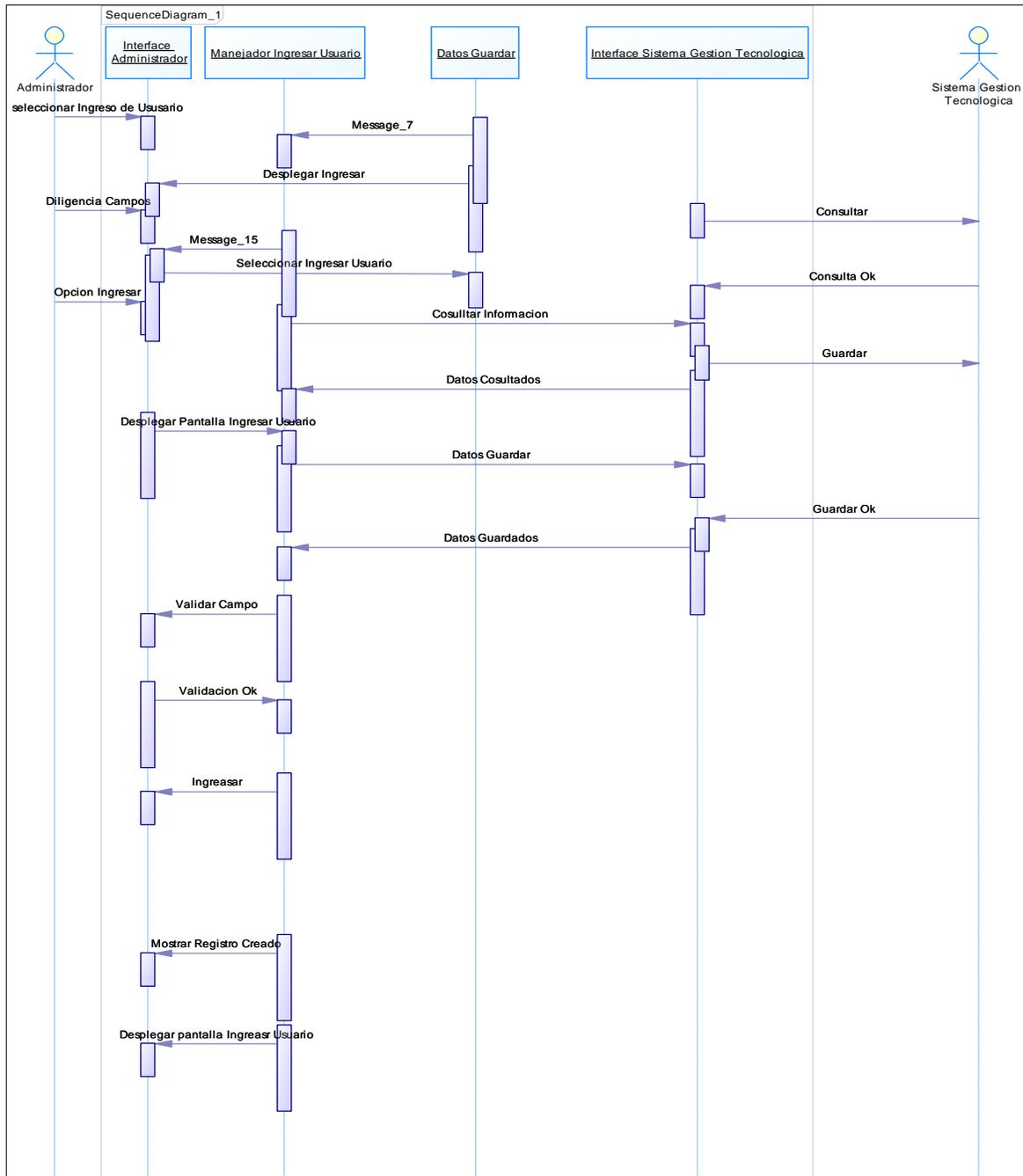
Figura 2. Diagrama selección equipo



Elaborado por: El autor

DIAGRAMA INGRESAR USUARIO

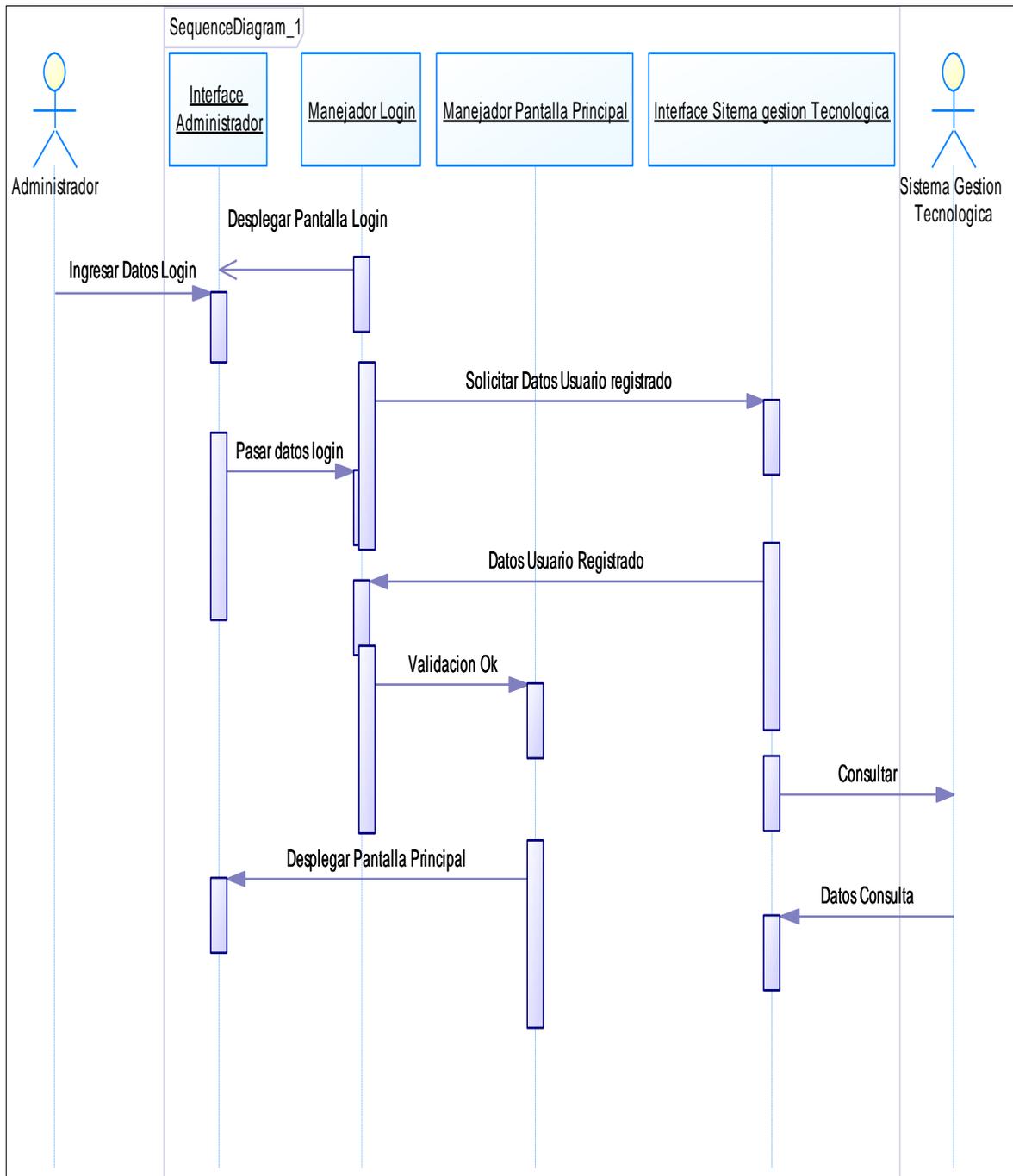
Figura 3. Diagrama ingresar usuario



Elaborado por: El autor

DIAGRAMA USO LOGIN

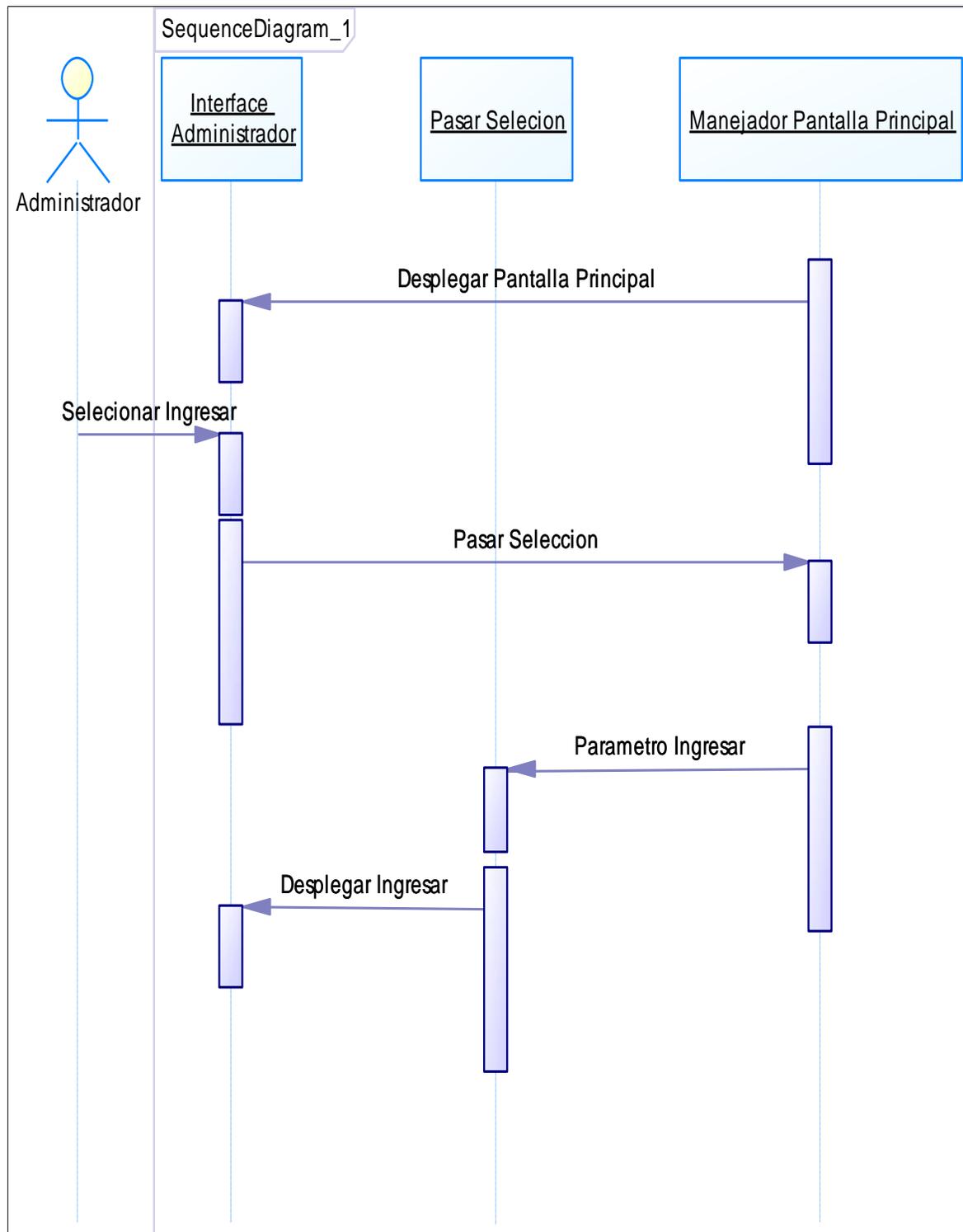
Figura 4. Diagrama Uso de Login



Elaborado por: El autor

DIAGRAMA INVENTARIO EQUIPO

Figura 5. Diagrama Inventario Equipo



Elaborado por: El autor

11.9. Arquitectura de desarrollo del software

11.9.1. Modelo Cliente/Servidor

Es un modelo basado en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios. Además esta relación está establecida en función del intercambio de mensajes que es el único elemento de acoplamiento entre ambos.

De estas líneas se deducen los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas Cliente/Servidor: el proceso cliente que es quien inicia el diálogo, el proceso servidor que pasivamente espera a que lleguen peticiones de servicio y el middleware que corresponde a la interfaz que provee la conectividad entre el cliente y el servidor para poder intercambiar mensajes.

11.9.2. Cliente

Es todo proceso que reclama servicios de otro. Una definición un poco más elaborada podría ser la siguiente: cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Se lo conoce con el término front-end.

Éste normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario
- Interactuar con el usuario
- Procesar la lógica de la aplicación y hacer validaciones locales
- Generar requerimientos de bases de datos
- Recibir resultados del servidor
- Formatear resultados

La funcionalidad del proceso cliente marca la operativa de la aplicación (flujo de información o lógica de negocio). De este modo el cliente se puede clasificar en:

Cliente basado en aplicación de usuario. Si los datos son de baja interacción y están fuertemente relacionados con la actividad de los usuarios de esos clientes.

Cliente basado en lógica de negocio. Toma datos suministrados por el usuario y/o la base de datos y efectúa los cálculos necesarios según los requerimientos del usuario.

11.9.3. Servidor

Es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se enumeran a continuación:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
 - Procesar requerimientos de bases de datos.
 - Formatear datos para transmitirlos a los clientes.
 - Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.
 - Puede darse el caso que un servidor actúe a su vez como cliente de otro servidor.
- Existen numerosos tipos de servidores, cada uno de los cuales da lugar a un tipo de arquitectura Cliente/Servidor diferente.

El término “servidor” se suele utilizar también para designar el hardware, de gran potencia, capacidad y prestaciones, utilizado para albergar servicios que atienden a un gran número de usuarios concurrentes. Desde el punto de vista de la arquitectura cliente/servidor y del procesamiento cooperativo un servidor es un servicio software que atiende las peticiones de procesos software clientes. (López Torralba, 2015).

11.9.4. Tipología del sistema

- **Multiplataforma**

El software puede ser instalado en Windows.

- **Multiusuario**

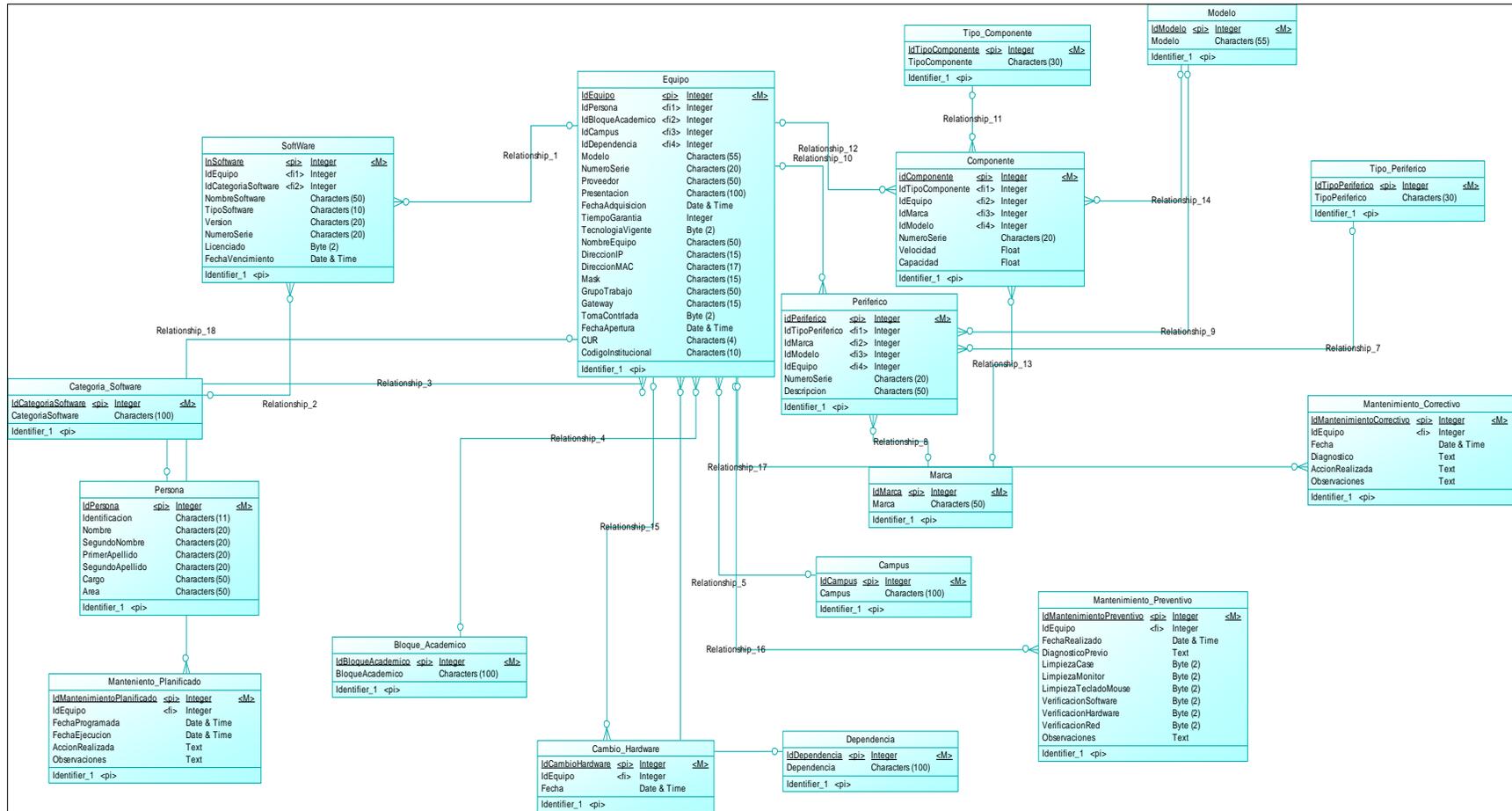
Varios usuarios acceder al sistema al mismo tiempo

- **Multitarea**

El sistema permite acceder a varias interfaces al mismo tiempo

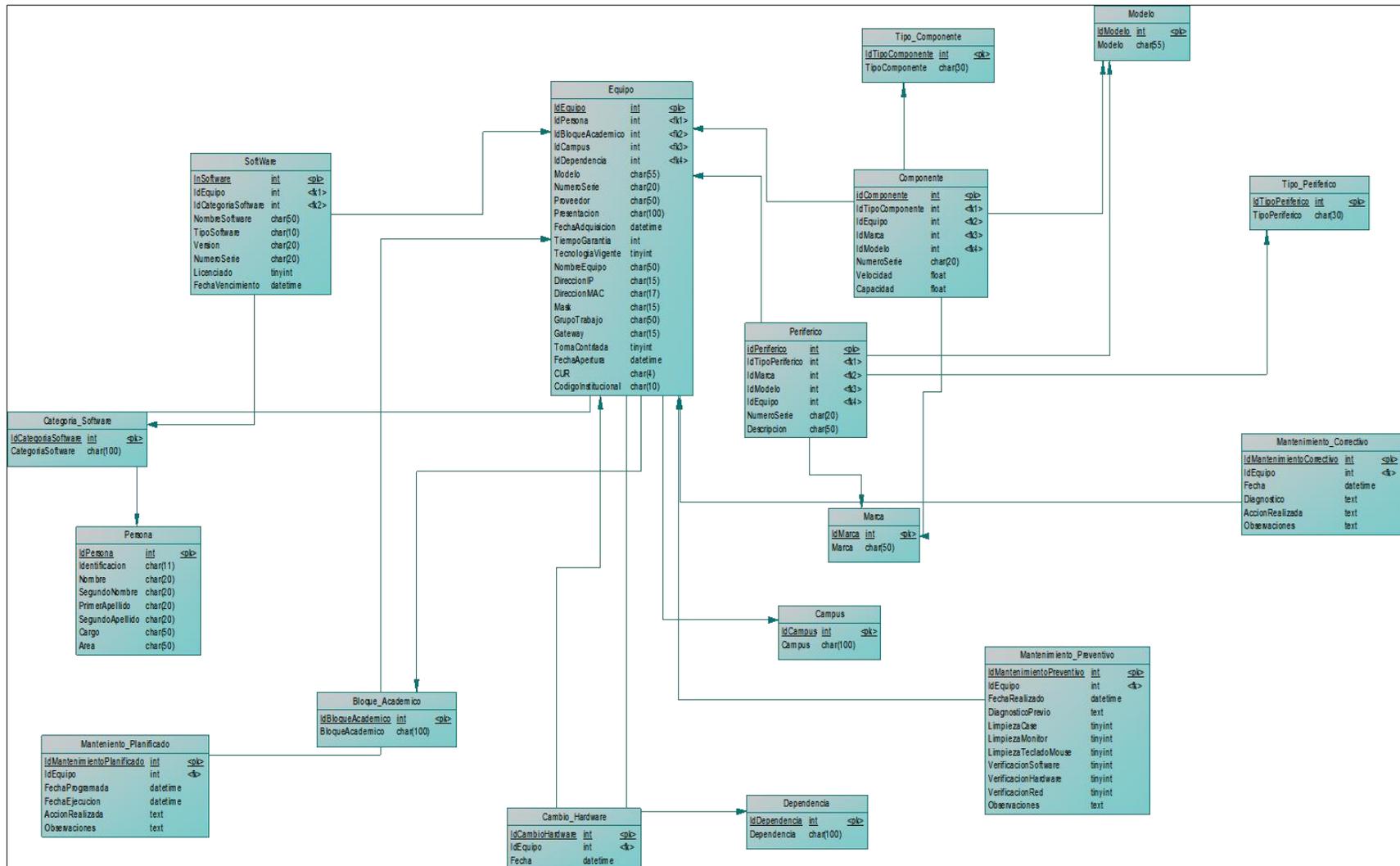
11.9.5. Modelos de las Base de Datos

Figura 6. Modelo Lógico



Elaborado por: El autor

Figura 7. Modelo físico



Elaborado por: El autor

11.9.6. Pruebas del sistema

En un proceso de pruebas formal, suelen confundirse con mucha facilidad, los niveles de pruebas con los tipos de prueba, y a pesar de que se encuentren íntimamente relacionadas, tienen connotaciones diferentes en el proceso. Para entender un poco más, vamos a partir del hecho de que las pruebas pueden ejecutarse en cualquier punto del proceso de desarrollo de software, y es aquí donde los niveles de prueba nos permiten entender con claridad los diferentes puntos o etapas en donde pueden ejecutarse ciertos tipos de prueba. Por lo anterior, es común que algunas personas se refieran a los niveles de pruebas o intenten clasificarlos como: pruebas de desarrollador, pruebas funcionales y pruebas de usuario final. Sin embargo, la terminología apropiada para referirse a los diferentes niveles corresponde a las siguientes cuatro (4) clasificaciones que son: pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación. En cada uno de estos niveles de prueba, se podrán ejecutar diferentes tipos de prueba tales como: pruebas funcionales, no funcionales, de arquitectura y asociadas al cambio de los productos.

Pruebas Unitarias o de Componente.- Son ejecutadas normalmente por el equipo de desarrollo, básicamente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada. Adicionalmente, Las pruebas sobre componentes unitarios, suelen denominarse pruebas de módulos o pruebas de clases, siendo la convención definida por el lenguaje de programación la que influye en el término a utilizar. Por último, es importante que toda la funcionalidad de cada componente unitario sea cubierta, por al menos, dos casos de prueba, los cuales deben centrarse en probar al menos una funcionalidad positiva y una negativa.

Pruebas de Integración.- Son monitoreadas por el equipo de desarrollo y consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta.

Pruebas de Sistema.- Son ejecutadas idealmente por un equipo de pruebas ajeno al equipo de desarrollo, una buena práctica en este punto corresponde a la tercerización de esta responsabilidad. La obligación de este equipo, consiste en la ejecución de actividades de prueba en donde se debe verificar que la funcionalidad total de un sistema fue implementada de acuerdo a los documentos de especificación definidos en el proyecto. Los casos de prueba a diseñar en este nivel de pruebas, deben cubrir los aspectos funcionales y no funcionales del sistema. Para el diseño de los casos de prueba en este nivel, el equipo debe utilizar como bases de prueba entregables tales como: requerimientos iniciales, casos de uso, historias de usuario, diseños, manuales técnicos y de usuario final, etc. Por último, es importante que los tipos de pruebas ejecutados en este nivel se desplieguen en un ambiente de pruebas / ambiente de pre-producción cuya infraestructura y arquitectura sea similar al ambiente de producción, evitando en todos los casos utilizar el ambiente real del cliente, debido principalmente, a que pueda ocasionar fallos en los servidores, lo que ocasionaría indisponibilidad en otros servicios alojados en este ambiente.

Pruebas de Aceptación.- Independientemente de que se haya tercerizado el proceso de pruebas y así la firma responsable de estas actividades haya emitido un certificado de calidad sobre el sistema objeto de prueba, es indispensable, que el cliente designe a personal que haga parte de los procesos de negocio para la ejecución de pruebas de aceptación, es incluso recomendable, que los usuarios finales que participen en este proceso, sean independientes al personal que apoyó el proceso de desarrollo. Cuando las pruebas de aceptación son ejecutadas en instalaciones o ambientes proporcionados por la firma desarrolladora se les denominan pruebas Alpha, cuando son ejecutadas desde la infraestructura del cliente se les denomina pruebas Beta. En los casos en que las pruebas de aceptación del producto se hayan ejecutado en el ambiente del proveedor, el aplicativo no podrá salir a producción, sin que se hayan ejecutados las respectivas pruebas Beta en el ambiente del cliente, de lo anterior es importante concluir, que las pruebas Alpha son opcionales, pero las pruebas Beta son obligatorias. (Zapata, 2013).

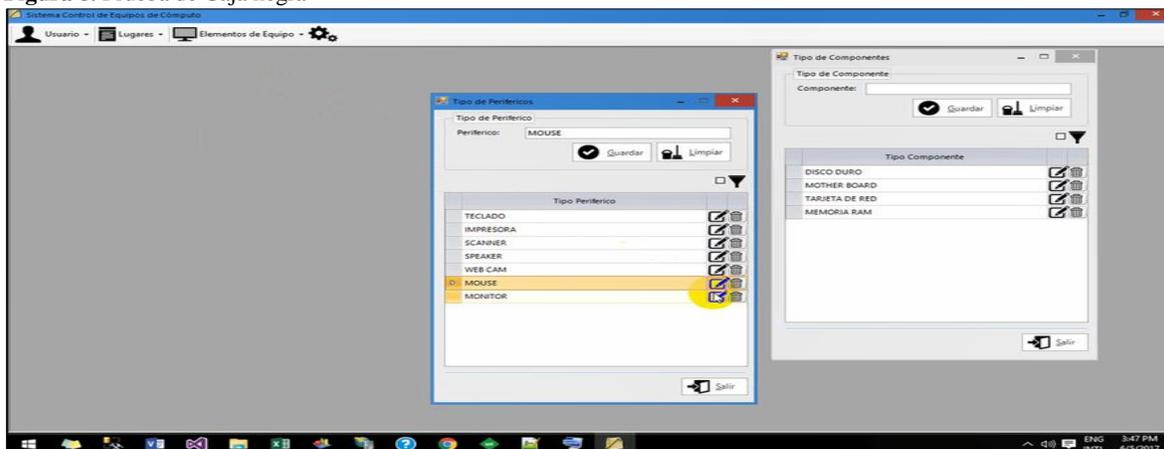
Técnicas de diseño de casos de prueba.- Permiten seleccionar la menor cantidad de casos con mayor probabilidad de encontrar fallas en el sistema. Por este motivo, estos casos se

consideran los más interesantes para ejecutar, ya que el testing exhaustivo no solo es imposible de ejecutar en un tiempo acotado, sino que también es muy caro e ineficiente.

Es así que se vuelve necesario seleccionar de una forma inteligente los valores de entrada que tengan más posibilidades de descubrir un error. Para esto, el diseño de pruebas se basa en técnicas bien conocidas y utilizadas, tales como particiones de equivalencia, valores límites, combinaciones por pares, tablas de decisión o máquinas de estado. Por ejemplo probar valores límites, basado en que hay siempre probabilidad de que no se estén controlando estas situaciones por errores al programar y utilizar un “menor” en lugar de un “menor igual”, etc.

Caja blanca y Caja negra.- La clasificación más importante y más difundida de técnicas de diseño de casos de prueba es basada en la información utilizada como entrada. Si utilizamos información interna del sistema que estamos probando, tal como el código, esquema de base de datos, etc., entonces se dice que estamos siguiendo una estrategia de caja blanca. Si en cambio nos basamos únicamente en la observación de entradas y salidas del sistema, estamos siguiendo un enfoque de caja negra.

Figura 8. Prueba de Caja negra



Efectivamente, ese caso sería como considerar que el sistema es una caja a la cual no podemos mirar su interior. Por esta asimilación es también que a las técnicas de caja blanca a veces se les dice técnicas de “caja transparente”, haciendo mejor referencia a que la idea es poder mirar qué hay dentro de esa caja que estamos probando. Podríamos decir que con

caja blanca nos preocupamos por lo que pasa dentro de la caja y con caja negra nos preocupamos por lo que pasa fuera de ella. Muchas veces el límite no está claro, o tal vez estamos siguiendo un enfoque de caja negra, pero como sabemos algo de lo que sucede dentro entonces aprovechamos esa información. También hay quienes hablan de “caja gris” cuando se combinan ambos enfoques.

Figura 9. Prueba de Caja blanca

Nombre	Marca	Modelo	Presentacion	Proveedor	Garantia
dell	DELL	INSPIRON 3523	LAPTOP	MERCADO NEGRO	2 Años
HP	KINGSTON	INSPIRON 3523	LAPTOP	sdsd	4 Años

11.9.7. Checklist de Aceptación de Pruebas

Mediante la aplicación del checklist se podrán evidenciar ciertas particularidades en la fase de pruebas del sistema, es necesario que se analice cada ítem con su respectiva verificación durante el proceso de pruebas.

CHECKLIST DE ACEPTACIÓN		
Título del Proyecto de Investigación		
DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ		
Pruebas	SI	NO
✓ ¿Se entregó el soporte solicitado por el personal de desarrollo?	✓	
✓ ¿Se realizaron las pruebas de la configuración funcional y física?	✓	
✓ ¿Se informó sobre el estado de avance de las actividades?	✓	
✓ ¿Se informaron sobre los errores detectados?	✓	
✓ ¿Se evaluó los requerimientos de software y su documentación?	✓	
Observaciones		
No existen observaciones debido que se realizó las debidas acciones con el pasoword que manejaría el sistema de gestión informático		

Elaborado por: El autor

11.9.8. Ventanas principales del sistema

Figura 10. Inventario de equipos de cómputo

Nombre	Marca	Modelo	Presentación	Proveedor	Garantía
dell	DELL	INSPIRON 3523	LAPTOP	MERCADO NEGRO	2 Años
HP	KINGSTON	INSPIRON 3523	LAPTOP	isdid	4 Años

Elaborado por: El autor

Figura 11. Equipos de cómputo

Datos Generales del Equipo

Nombre: dell Ficha No: 6 Marca: DELL Proveedor: MERCADO NEGRO
 Número: 0 Campus: CAMPUS1 Modelo: INSPIRON 3523 Fecha Adquisición: 13/6/2017
 Fecha Apertura: 13/6/2017 Bloque Acad: BLOQUE 1 No. Serie: 259-DSD233 Presentación: LAPTOP
 Código Institucional: 0000 Dependencia: LA MANA CUR: 123 Garantía (Años):

Guardar Cancelar

Perifericos Componentes Softwares Conectividad Usuarios

Componente	Marca	Modelo	Numero Serie	Velocidad...	Capacidad...	Tipo
MOTHERBOARD	DELL		isdudifug	0	0	

Datos de Componentes

Componente: Seleccione Marca: Seleccione Modelo: Seleccione No. Serie: Velocidad: 0.00 HGz Capacidad: 0.00 GB Tipo:

Guardar Cancelar

Elaborado por: El autor

Figura 12. Datos generales del equipo

Sistema Control de Equipos de Cómputo

Equipos de Cómputo

Datos Generales del Equipo

Nombre: dell Ficha No: 6 Marca: DELL Proveedor: MERCADO NEGRO

Número: 0 Campus: CAMPUS1 Modelo: INSPIRON 3523 Fecha Adquisición: 13/6/2017

Fecha Apertura: 13/6/2017 Bloque Acad: BLOQUE 1 No. Serie: 259-DSD233 Presentación: LAPTOP

Codigo Institucional: 0000 Dependencia: LA MANA CUR: 123 Garantía (Años): 2

Guardar Cancelar

Perifericos Componentes Softwares Conectividad Usuarios

Periferico	Marca	Modelo	Numero Serie	Descripcion
Teclado	DELL	INSPIRON 3523	vdfgd	gdfgd
MOUSE	DELL	INSPIRON 3523	456y-523	rgf

Datos de Periferico

Periferico: Seleccione

Marca: Seleccione

Modelo: Seleccione

No. Serie:

Descripción:

Guardar Cancelar

Elaborado por: El autor

Figura 13. Cambio de Periféricos

Sistema Control de Equipos de Cómputo

Equipos de Cómputo

Datos Generales del Equipo

Nombre: dell Ficha No: 6 Marca: DELL Proveedor: MERCADO NEGRO

Número: 0 Campus: CAMPUS1 Modelo: INSPIRON 3523 Fecha Adquisición: 13/6/2017

Fecha Apertura: 13/6/2017 Bloque Acad: BLOQUE 1 No. Serie: 259-DSD233 Presentación: LAPTOP

Codigo Institucional: 0000 Dependencia: LA MANA CUR: 123 Garantía (Años): 2

Guardar Cancelar

Perifericos Componentes Softwares Conectividad Usuarios

Periferico: Seleccione

Marca: Seleccione

Modelo: Seleccione

No. Serie:

Descripción:

Guardar Cancelar

Confirma que desea realizar la operación?

Se desea cambiar el periferico con numero de serie VDFGD haga click en Aceptar y seguidamente inserte el nuevo periferico.

Aceptar Cancelar

Elaborado por: El autor

Figura 14. Cambio de Hardware

Sistema Control de Equipos de Cómputo

Equipos de Cómputo

Datos Generales del Equipo

Nombre: dell Ficha No: 6 Marca: DELL Proveedor: MERCADO NEGRO
 Número: 0 Campus: CAMPUS1 Modelo: INSPIRON 3523 Fecha Adquisición: 13/6/2017
 Fecha Apertura: 13/6/2017 Bloque Acad: BLOQUE 1 No. Serie: 259-OSD233 Presentación: LAPTOP
 Código Institucional: 0000 Dependencia: LA MANA CUR: 123 Garantía (Años): 2

Guardar Cancelar

Perifericos Componentes Softwares Conectividad Usuarios

Periferico	Marca	Modelo	Numero Serie	Descripción
MOUSE	DELL	INSPIRON 3523	456y-523	rgf
Teclado	KINGSTON	LOG120	dfdf	dfhfh

Datos de Periferico

Periferico: Seleccione
 Marca: Seleccione
 Modelo: Seleccione
 No. Serie:
 Descripción:

Guardar Cancelar

Cambio de Hardware

Datos del Cambio

Fecha del Cambio: 23/6/2017
 Descripción: ksndisg

Guardar

Elaborado por: El autor

Figura 15. Mantenimiento

Sistema Control de Equipos de Cómputo

Inventario de Equipos de Cómputo

Nuevo Modificar Imprimir Baja Técnica Mantenimientos Ver Filtros

Nombre	Marca	Modelo	Presentación	Proveedor	Garantía
--------	-------	--------	--------------	-----------	----------

Mantenimientos

Mtto. Preventivo Mttto. Correctivo Mttto. Planificado

Fecha Realizado: Fecha Diagnostico Previo Limp Case Limp Monitor Limp Teclado o Mouse Verif Software Verif Hardware Verif Red Observaciones

Diagnostico Previo:

Observaciones:

Limpieza externa - interna

Case: SI NO

Monitor: SI NO

Teclado y Mouse: SI NO

Funcionalidad

Verificación de software: SI NO

Verificación de hardware: SI NO

Verificación de red: SI NO

Guardar Cancelar

Elaborado por: El autor

Figura 16. Historial del Equipo

The screenshot shows the 'Historial de Equipo' window in the 'Sistema Control de Equipos de Computo' application. The main table lists equipment with the following data:

No. Equipo	Nombre	Marca	Modelo	Presentacion	Proveedor	Garantia
1	dell	DELL	INSPIRON 3523	LAPTOP	MERCADO NEGRO	2 Años
2	HP	KINGSTON	INSPIRON 3523	LAPTOP	sdds	4 Años

Below the main table, there are two sections for peripherals:

Perifericos

Periferico	Fecha	Descripcion	Numero Serie a Reemplazar	Numero Serie Nuevo
MOUSE	23/6/2017 18:42...		gh	gk
Teclado	23/6/2017 16:32...	hsthdthdf	23	wrgrg
Teclado	23/6/2017 18:55...		wrgrg	Jygh7
MOUSE	23/6/2017 16:32...	se cambio porque estaba roto	gk	456y-523
Teclado	23/6/2017 16:32...	ksdndsg	vdtdg	dtftd

Perifericos Removidos

Periferico	Marca	Modelo	Numero Serie	Descripcion
Teclado	DELL	INSPIRON 3523	cc	fgf
MOUSE	vbn	LOG120	6ytt6	
Teclado	KINGSTON	LOG120	6666666	lkkkkkkkkkk
Teclado	DELL	LOG120	trurtr	urjurt
Teclado	KINGSTON	LOG120	5yutr	utrur
MOUSE	xbv	LOG120	gh	ff
Teclado	KINGSTON	INSPIRON 3523	23	45

Elaborado por: El autor

11.9.9. Herramientas de desarrollo de la propuesta

Es importante mencionar que para el desarrollo del sistema se utilizaron herramientas las cuales se integran adecuadamente en la consecución de la propuesta planteada:

11.10. Motor de base de datos:

11.10.1. SQL Server 2008

Es el sistema de bases de datos profesional de Microsoft. Contiene una variedad de características y herramientas que se pueden utilizar para desarrollar y administrar bases de datos y soluciones de todo tipo basadas en ellas. El motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos. Así mismo, proporciona acceso controlado y procesamiento rápido de transacciones para cumplir los requisitos de las aplicaciones de base de datos más exigentes. Dependiendo del Motor de base de datos tenemos los componentes Service Broker y Replicación.

Service Broker.- Ayuda a los desarrolladores de software a crear aplicaciones de base de datos escalables y seguras, esta nueva tecnología del Motor de Base de Datos proporciona

una plataforma de comunicación basada en mensajes que permite a los componentes de aplicación independientes trabajar como un conjunto funcional.

12. IMPACTOS TÉCNICOS, SOCIALES, AMBIENTALES O ECONÓMICOS

Técnico.- Una vez finalizado el presente proyecto de investigación es preciso señalar que el impacto es positivo pues el mismo permitirá un adecuado manejo de los equipos informáticos con los que cuenta la institución.

Social.- Es importante que todos los sistemas brinden alternativas que faciliten llevar los procesos de las organizaciones de manera ágil y precisa y que estos beneficien a determinados sectores de la sociedad.

Económico.- El sistema implementado permitirá desarrollar las tareas de registro de información de manera adecuada y sin contratiempos, dichos procesos sistematizados brindarán la posibilidad de optimizar recursos.

Ambiental.- Contribuir con el medio ambiente debe ser el papel esencial de los nuevos sistemas informáticos quienes sean capaces de generar armonía con la naturaleza al reducir el uso del papel.

13. PRESUPUESTO PARA EL DESARROLLO DE LA INVESTIGACIÓN

Tabla 14. Presupuesto

Resultados/Actividades	Primer año			
	1er Trimestre	2do Trimestre	3er Trimestre	4to Trimestre
Integración del grupo de trabajo				
Identificar requerimientos	\$200,00			
Diseño Metodológico	\$100,00	\$100,00	\$100,00	\$100,00
Establecer las herramientas para el desarrollo		\$50,00		
Ms SQL Server 2008			\$600,00	
Visual Estudios 2010			\$600,00	
Diseño del software		\$300,00	\$300,00	
Pruebas			\$50,00	\$50,00
Implementación				\$50,00
Proceso de capacitación al administrador				\$50,00
Subtotal:	\$300,00	\$450,00	\$1.650,00	\$250,00
TOTAL:	\$2.650,00			

Elaborado por: El autor

14. CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Es necesario partir de las necesidades identificadas en la institución como es la de contar con un sistema que permita gestionar la información adecuadamente.
- Como metodología de desarrollo se aplicó la metodología Ágil Scrum debido a que esta permite desarrollar software de calidad.
- Las herramientas utilizadas para el diseño del sistema deben prestar las mejores garantías de integración.

RECOMENDACIONES

- Es importante identificar de manera adecuada cada uno de los requerimientos y realizar su respectivo análisis con el fin de evitar errores.
- Se recomienda contar con un análisis de las metodologías de desarrollo de software y seleccionar aquella que preste las garantías efectivas de desarrollo.
- Seleccionar las herramientas adecuadas que cumplan las exigencias de desarrollo.

15. BIBLIOGRAFÍA

- Bueno, D. (2011). Tecnología de Sistemas de Control. Madrid: Editorial Obelisco, Segunda edición, 240 págs., ISBN: 658-643-1529.
- Cegarra, J. (2012). Los Métodos de Investigación. Madrid: Editorial Díaz de Santos, Primera edición, 180 págs., ISBN: 648-643-1029.
- Fernández, E., (2012). Introducción a Extreme Programming Ingeniería del Software. Madrid: RAMA Editorial, 250 págs., ISBN: 176-529-7230.
- Leyva, E., Prieto, J. (2012). Sistemas y Aplicaciones Informáticas. Madrid: Editorial MAD, Primera edición, 210 págs., ISBN: 921-723-521-32.
- Patrick, T., (2010). Visual Estudio .Net 2010, Madrid: Editorial Anaya Multimedia, Primera edición, 230 págs., ISBN: 735-123-642-903.
- Pérez, M., (2008). Motor de base de datos y administración. Madrid: Editorial RC Libros, 256 págs., ISBN: 256-167-12-087.
- Ramos, D., Noriega, R., Laínez, R. (2017). Curso de Ingeniería de Software, Madrid, IT Campus, 310 págs., ISBN: 547-109-76-091.
- Rivera, V., (2012), Diseño y modelado de software, Madrid, Editorial RC Libros, 235 págs., ISBN: 567-812-426-78.
- Terán D., (2014). Administración Estratégica de la función informática. México: Editorial Alfa omega, 120 págs., ISBN: 674-328-546-89.
- Zabala, J. (2011), Fundamentos de Cliente Servidor. Barcelona: Editorial RanMa, 215 págs., ISBN: 967-912-17-1042.
- Zapata, J. (2013). Ingeniería de Software con énfasis en pruebas, Buenos Aires, Editorial Dunken, 317 págs., ISBN: 178-326-3167-09.

- Zarzar, C. (2015). Métodos y pensamiento crítico, México: Editorial Patria, Primera edición, 190 págs., ISBN: 253-178-629-104.

Linkografía

- Cisneros, S., (2015). Centro de Investigaciones Sociológicas, Ref. [02/11/2015], Consulta [19/06/2017], disponible: http://www.cis.es/cis/opencms/ES/1_encuestas/ComoSeHacen/
- Lopez, T., (2015). Developer Network. Obtenido de Developer Network: Ref. [05/09/2015], Consulta [12/05/2017], disponible: [https://msdn.microsoft.com/es-es/library/fx6bk1f4\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/fx6bk1f4(v=vs.100).aspx)

16. ANEXOS

ANEXO 1. HOJAS DE VIDA DEL EQUIPO DE TRABAJO

CURRICULUM VITAE

NOMBRE:	Rufino Yimael	
PELLIDOS:	Chamaidan Panchana	
FECHA Y LUGAR DE NACIMIENTO:	27 de Enero de 1983	
EDAD:	34 AÑOS	
CEDULA DE IDENTIDAD:	0704006873	
LICENCIA DE CONDUCIR:	TIPO A Y TIPO B	
DIRECCIÓN	La Maná, Calle Pastaza y Saquisilí	
TELÉFONO	032689767 / 0994650669	
E-MAIL	rufino.chamaidan3@utc.edu.ec	

ESTUDIOS REALIZADOS

ESTUDIOS PRIMARIOS:

- Escuela Gnral.Manuel Serrano (El Guabo-El Oro)

ESTUDIOS SECUNDARIOS:

- Instituto Técnico Superior Dr. José Ochoa León”(Pasaje)

ESTUDIOS UNIVERSITARIOS:

- Ingeniería en Informática y Sistemas Computacionales

IDIOMAS:

- Certificado de Suficiencia en Inglés

CURRICULUM VITAE

DATOS PERSONALES



NOMBRES: Henry Mauricio
APELLIDOS: Chanatasig Toapanta
FECHA Y LUGAR DE NACIMIENTO: Saquisilí, Nov. 30 de 1982
EDAD: 34 años
NACIONALIDAD: ecuatoriano
CEDULA DE IDENTIDAD: 0502817646
IDENTIDAD MILITAR: 8205104103
LICENCIA DE CONCUCIR: Tipo B
DIRECCION RESIDENCIAL: Barrio Unión Panamericano
TELEFONOS:
CONVENCIONAL: 03-2722512
CELULAR: 0995789184
E-mail: henry.chanatasig@utc.edu.

OBJETIVO: Crear aplicaciones aplicando todos los pasos en lo que se refiere a Ingeniería de Software, desde requisitos hasta pruebas de software.

ESTUDIOS REALIZADOS

SUPERIOR: Universidad de las Fuerzas Armadas-ESPE
 Magister en Ingeniería de Software, Quito
 Septiembre 2015

SUPERIOR: Universidad Técnica de Cotopaxi
 Ingeniero en Informática y Sistemas
 Computacionales, Latacunga Diciembre 2009

SECUNDARIA: Colegio Nacional Saquisilí
 Bachiller en Ciencias Físico Matemático,
 Saquisilí Agosto 2001

PRIMARIA: Escuela Naciones Unidas-Saquisilí Julio 1995

IDIOMAS: Certificación Suficiencia en el Idioma Inglés

ANEXO 2. ENCUESTAS APLICADAS AL PERSONAL DOCENTE Y ADMINISTRATIVO DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ.

**UNIVERSIDAD TÉCNICA DE COTOPAXI
EXTENSIÓN LA MANÁ**



ENCUESTA DIRIGIDA A LOS DOCENTES

PROYECTO: “DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI EXTENSIÓN LA MANÁ”

1. ¿Tiene usted conocimiento de cómo se lleva el registro de los equipos informáticos en el Departamento de Sistemas de Información?

SI

NO

2. ¿Conoce usted si existen problemas al momento de registrar el préstamo de los equipos en el Departamento de Sistemas de Información?

SI

NO

3. ¿Cree usted que el registro de la información de los equipos de cómputo es el adecuado?

SI

NO

4. ¿Conoce usted sobre algún sistema el que permita llevar un registro adecuado sobre los equipos en el departamento de sistemas de información?

SI NO

5. ¿Supone usted que sería viable la implementación de un sistema para la gestión de los equipos en el Departamento de Sistemas de Información?

SI NO

6. ¿Cree usted que la implementación de un sistema para la gestión de los equipos en el Departamento de Sistemas de Información permita establecer una mejor organización de la información?

SI NO

7. Está usted de acuerdo que este sistema ayudará a la Institución a organizar la gestión de los equipos en el Departamento de Sistemas de Información?

SI NO

Anexo 3. Script de la base de datos

```

USE [InventarioPC]
GO
/***** Object: Table [dbo].[Bloque_Academico]    Script Date: 11/7/2017 10:41:04
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Bloque_Academico](
    [IdBloqueAcademico] [int] IDENTITY(1,1) NOT NULL,
    [BloqueAcademico] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_BLOQUE_ACADEMICO] PRIMARY KEY CLUSTERED
(
    [IdBloqueAcademico] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Cambio_Periferico]    Script Date: 11/7/2017 10:41:05
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Cambio_Periferico](
    [IdCambioPeriferico] [int] IDENTITY(1,1) NOT NULL,
    [IdEquipo] [int] NOT NULL,
    [TipoPeriferico] [nvarchar](30) NOT NULL,
    [Fecha] [datetime] NOT NULL,
    [Descripcion] [nvarchar](max) NULL,
    [NumeroSerieNuevo] [nvarchar](50) NOT NULL,
    [NumeroSerieViejo] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_CAMBIO_PERIFERICO] PRIMARY KEY CLUSTERED
(
    [IdCambioPeriferico] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/***** Object: Table [dbo].[Campus]    Script Date: 11/7/2017 10:41:05 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Campus](
    [IdCampus] [int] IDENTITY(1,1) NOT NULL,
    [Campus] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_CAMPUS] PRIMARY KEY CLUSTERED
(
    [IdCampus] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Categoria_Software]    Script Date: 11/7/2017 10:41:05
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Categoria_Software](
    [IdCategoriaSoftware] [int] IDENTITY(1,1) NOT NULL,
    [CategoriaSoftware] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_CATEGORIA_SOFTWARE] PRIMARY KEY CLUSTERED
(
    [IdCategoriaSoftware] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Componente]    Script Date: 11/7/2017 10:41:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Componente](
    [IdComponente] [int] IDENTITY(1,1) NOT NULL,
    [IdEquipo] [int] NOT NULL,
    [TipoComponente] [nvarchar](30) NOT NULL,
    [IdMarca] [int] NOT NULL,
    [IdModelo] [int] NULL,
    [NumeroSerie] [nvarchar](50) NULL,
    [Velocidad] [decimal](13, 2) NULL,

```

```

    [Capacidad] [decimal](13, 2) NULL,
    [Tipo] [nvarchar](20) NULL,
    CONSTRAINT [PK_COMPONENTE] PRIMARY KEY CLUSTERED
(
    [IdComponente] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

```

/***** Object: Table [dbo].[Conectividad]  Script Date: 11/7/2017 10:41:05 *****/
SET ANSI_NULLS ON

```

GO

```

SET QUOTED_IDENTIFIER ON

```

GO

```

CREATE TABLE [dbo].[Conectividad](
    [IdConectividad] [int] IDENTITY(1,1) NOT NULL,
    [IdEquipo] [int] NOT NULL,
    [DireccionIP] [nvarchar](15) NOT NULL,
    [DireccionMAC] [nvarchar](17) NULL,
    [Mask] [nvarchar](15) NULL,
    [GrupoTrabajo] [nvarchar](50) NULL,
    [Gateway] [nvarchar](15) NULL,
    [TomaControlada] [bit] NULL,
    CONSTRAINT [PK_CONECTIVIDAD] PRIMARY KEY CLUSTERED
(
    [IdConectividad] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

```

/***** Object: Table [dbo].[Dependencia]  Script Date: 11/7/2017 10:41:05 *****/
SET ANSI_NULLS ON

```

GO

```

SET QUOTED_IDENTIFIER ON

```

GO

```

CREATE TABLE [dbo].[Dependencia](
    [IdDependencia] [int] IDENTITY(1,1) NOT NULL,
    [Dependencia] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_DEPENDENCIA] PRIMARY KEY CLUSTERED
(
    [IdDependencia] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

```
/***** Object: Table [dbo].[Equipo] Script Date: 11/7/2017 10:41:05 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[Equipo](
    [IdEquipo] [int] IDENTITY(1,1) NOT NULL,
    [IdBloqueAcademico] [int] NOT NULL,
    [IdCampus] [int] NOT NULL,
    [IdDependencia] [int] NOT NULL,
    [IdMarca] [int] NOT NULL,
    [IdModelo] [int] NOT NULL,
    [NombreEquipo] [nvarchar](50) NOT NULL,
    [NumeroSerie] [nvarchar](50) NULL,
    [Proveedor] [nvarchar](50) NULL,
    [Presentacion] [nvarchar](100) NULL,
    [FechaAdquisicion] [datetime] NULL,
    [TiempoGarantia] [int] NULL,
    [TecnologiaVigente] [bit] NULL,
    [FechaApertura] [datetime] NULL,
    [CUR] [nvarchar](4) NULL,
    [CodigoInstitucional] [nvarchar](10) NULL,
    [Tipo] [nvarchar](10) NULL,
    [NoFicha] [nvarchar](10) NULL,
    [NumeroEquipo] [int] NULL,
    CONSTRAINT [PK_EQUIPO] PRIMARY KEY CLUSTERED
(
```

```
    [IdEquipo] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

```
/***** Object: Table [dbo].[Mantenimiento_Correctivo] Script Date: 11/7/2017
10:41:05 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[Mantenimiento_Correctivo](
```

```

        [IdMantenimientoCorrectivo] [int] IDENTITY(1,1) NOT NULL,
        [IdEquipo] [int] NULL,
        [Fecha] [datetime] NOT NULL,
        [Diagnostico] [nvarchar](max) NULL,
        [AccionRealizada] [nvarchar](max) NULL,
        [Observaciones] [nvarchar](max) NULL,
    CONSTRAINT [PK_MANTENIMIENTO_CORRECTIVO] PRIMARY KEY
    CLUSTERED
    (
        [IdMantenimientoCorrectivo] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
    ON) ON [PRIMARY]
    ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

```

GO

```

/***** Object: Table [dbo].[Mantenimiento_Planificado]    Script Date: 11/7/2017
10:41:05 *****/

```

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```

CREATE TABLE [dbo].[Mantenimiento_Planificado](
    [IdMantenimientoPlanificado] [int] IDENTITY(1,1) NOT NULL,
    [IdEquipo] [int] NULL,
    [FechaProgramada] [datetime] NOT NULL,
    [FechaEjecucion] [datetime] NOT NULL,
    [AccionRealizada] [nvarchar](max) NULL,
    [Observaciones] [nvarchar](max) NULL,
    CONSTRAINT [PK_MANTENIMIENTO_PLANIFICADO] PRIMARY KEY
    CLUSTERED
    (
        [IdMantenimientoPlanificado] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
    ON) ON [PRIMARY]
    ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

```

GO

```

/***** Object: Table [dbo].[Mantenimiento_Preventivo]    Script Date: 11/7/2017
10:41:05 *****/

```

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```

CREATE TABLE [dbo].[Mantenimiento_Preventivo](
    [IdMantenimientoPreventivo] [int] IDENTITY(1,1) NOT NULL,

```

```

        [IdEquipo] [int] NULL,
        [FechaRealizado] [datetime] NOT NULL,
        [DiagnosticoPrevio] [nvarchar](max) NULL,
        [LimpiezaCase] [bit] NULL,
        [LimpiezaMonitor] [bit] NULL,
        [LimpiezaTecladoMouse] [bit] NULL,
        [VerificacionSoftware] [bit] NULL,
        [VerificacionHardware] [bit] NULL,
        [VerificacionRed] [bit] NULL,
        [Observaciones] [nvarchar](max) NULL,
    CONSTRAINT [PK_MANTENIMIENTO_PREVENTIVO] PRIMARY KEY
    CLUSTERED
    (
        [IdMantenimientoPreventivo] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
    ON) ON [PRIMARY]
    ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/***** Object: Table [dbo].[Marca] Script Date: 11/7/2017 10:41:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Marca](
        [IdMarca] [int] IDENTITY(1,1) NOT NULL,
        [Marca] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_MARCA] PRIMARY KEY CLUSTERED
    (
        [IdMarca] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
    ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Modelo] Script Date: 11/7/2017 10:41:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Modelo](
        [IdModelo] [int] IDENTITY(1,1) NOT NULL,
        [Modelo] [nvarchar](55) NOT NULL,
    CONSTRAINT [PK_MODELO] PRIMARY KEY CLUSTERED
    (

```

```

        [IdModelo] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
    ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

```

/***** Object: Table [dbo].[Periferico] Script Date: 11/7/2017 10:41:05 *****/
SET ANSI_NULLS ON

```

GO

```

SET QUOTED_IDENTIFIER ON

```

GO

```

CREATE TABLE [dbo].[Periferico](
    [IdPeriferico] [int] IDENTITY(1,1) NOT NULL,
    [IdEquipo] [int] NOT NULL,
    [IdMarca] [int] NOT NULL,
    [IdModelo] [int] NOT NULL,
    [TipoPeriferico] [nvarchar](30) NOT NULL,
    [NumeroSerie] [nvarchar](max) NOT NULL,
    [Descripcion] [nvarchar](50) NULL,
    [Activo] [bit] NOT NULL,
    CONSTRAINT [PK_PERIFERICO] PRIMARY KEY CLUSTERED
(
    [IdPeriferico] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
    ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

```

GO

```

/***** Object: Table [dbo].[Persona] Script Date: 11/7/2017 10:41:05 *****/
SET ANSI_NULLS ON

```

GO

```

SET QUOTED_IDENTIFIER ON

```

GO

```

CREATE TABLE [dbo].[Persona](
    [IdPersona] [int] IDENTITY(1,1) NOT NULL,
    [IdEquipo] [int] NOT NULL,
    [Identificacion] [nvarchar](11) NULL,
    [Nombre] [nvarchar](20) NOT NULL,
    [PrimerApellido] [nvarchar](20) NOT NULL,
    [SegundoApellido] [nvarchar](20) NULL,
    [Cargo] [nvarchar](50) NULL,
    [Area] [nvarchar](50) NULL,
    CONSTRAINT [PK_PERSONA] PRIMARY KEY CLUSTERED
(
    [IdPersona] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

```
/****** Object: Table [dbo].[Seg_Funcion] Script Date: 11/7/2017 10:41:05 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[Seg_Funcion](
    [IdFuncion] [int] IDENTITY(1,1) NOT NULL,
    [Funcion] [nvarchar](30) NOT NULL,
    [TagIdentifier] [nvarchar](50) NULL,
    CONSTRAINT [PK_SEG_FUNCION] PRIMARY KEY CLUSTERED
```

(

```
    [IdFuncion] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

GO

```
/****** Object: Table [dbo].[Seg_Login] Script Date: 11/7/2017 10:41:05 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[Seg_Login](
    [IdLogin] [int] IDENTITY(1,1) NOT NULL,
    [IdRol] [int] NOT NULL,
    [Usuario] [nvarchar](20) NOT NULL,
    [Clave] [nvarchar](200) NOT NULL,
    [Estado] [bit] NULL,
    [Intento] [int] NULL,
    [Logueado] [nvarchar](20) NULL,
    CONSTRAINT [PK_SEG_LOGIN] PRIMARY KEY CLUSTERED
```

(

```
    [IdLogin] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

GO

```
/****** Object: Table [dbo].[Seg_Permiso] Script Date: 11/7/2017 10:41:05 *****/
```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Seg_Permissio](
    [IdPermiso] [int] IDENTITY(1,1) NOT NULL,
    [IdRol] [int] NOT NULL,
    [Nuevo] [bit] NULL,
    [Edicion] [bit] NULL,
    [Eliminacion] [bit] NULL,
    [Reporte] [bit] NULL,
    CONSTRAINT [PK_SEG_PERMISO] PRIMARY KEY CLUSTERED
(
    [IdPermiso] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
/***** Object: Table [dbo].[Seg_Rol]   Script Date: 11/7/2017 10:41:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Seg_Rol](
    [IdRol] [int] IDENTITY(1,1) NOT NULL,
    [Rol] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_SEG_ROL] PRIMARY KEY CLUSTERED
(
    [IdRol] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
/***** Object: Table [dbo].[Seg_Rol_Funcion]   Script Date: 11/7/2017 10:41:05
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Seg_Rol_Funcion](
    [IdFuncion] [int] NOT NULL,
    [IdRol] [int] NOT NULL,
    CONSTRAINT [PK_SEG_ROL_FUNCION] PRIMARY KEY CLUSTERED

```

```
(
    [IdFuncion] ASC,
    [IdRol] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

/***** Object: Table [dbo].[Seg_Traza] Script Date: 11/7/2017 10:41:05 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[Seg_Traza](
    [IdTraza] [int] IDENTITY(1,1) NOT NULL,
    [Fecha] [datetime] NULL,
    [Usuario] [nvarchar](20) NULL,
    [Accion] [nvarchar](50) NULL,
    [Tabla] [nvarchar](20) NULL,
    CONSTRAINT [PK_SEG_TRAZA] PRIMARY KEY CLUSTERED
```

```
(
    [IdTraza] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

/***** Object: Table [dbo].[SoftWare] Script Date: 11/7/2017 10:41:05 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[SoftWare](
    [IdSoftware] [int] IDENTITY(1,1) NOT NULL,
    [IdEquipo] [int] NOT NULL,
    [IdCategoriaSoftware] [int] NOT NULL,
    [NombreSoftware] [nvarchar](50) NOT NULL,
    [TipoSoftware] [nvarchar](15) NULL,
    [Version] [nvarchar](20) NULL,
    [NumeroSerie] [nvarchar](20) NULL,
    [Licenciado] [bit] NULL,
    [FechaVencimiento] [nvarchar](15) NULL,
    CONSTRAINT [PK_SOFTWARE] PRIMARY KEY CLUSTERED
```

```
(
    [IdSoftware] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

```
/****** Object: Table [dbo].[Sys_SecurityKey] Script Date: 11/7/2017 10:41:05
*****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[Sys_SecurityKey](
    [_Key] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

GO

```
/****** Object: Table [dbo].[Tipo_Componente] Script Date: 11/7/2017 10:41:05
*****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[Tipo_Componente](
    [IdTipoComponente] [int] IDENTITY(1,1) NOT NULL,
    [TipoComponente] [nvarchar](30) NOT NULL,
    CONSTRAINT [PK_TIPO_COMPONENTE] PRIMARY KEY CLUSTERED
```

(

```
    [IdTipoComponente] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

GO

```
/****** Object: Table [dbo].[Tipo_Periferico] Script Date: 11/7/2017 10:41:05 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[Tipo_Periferico](
    [IdTipoPeriferico] [int] IDENTITY(1,1) NOT NULL,
    [TipoPeriferico] [nvarchar](30) NOT NULL,
    CONSTRAINT [PK_TIPO_PERIFERICO] PRIMARY KEY CLUSTERED
```

(

```
    [IdTipoPeriferico] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
```

```
ALTER TABLE [dbo].[Cambio_Periferico] WITH CHECK ADD CONSTRAINT
[FK_CAMBIO_P_REFERENCE_EQUIPO] FOREIGN KEY([IdEquipo])
REFERENCES [dbo].[Equipo] ([IdEquipo])
```

```
GO
```

```
ALTER TABLE [dbo].[Cambio_Periferico] CHECK CONSTRAINT
[FK_CAMBIO_P_REFERENCE_EQUIPO]
```

```
GO
```

```
ALTER TABLE [dbo].[Componente] WITH CHECK ADD CONSTRAINT
[FK_COMPONEN_COMPONENT_MARCA] FOREIGN KEY([IdMarca])
REFERENCES [dbo].[Marca] ([IdMarca])
```

```
GO
```

```
ALTER TABLE [dbo].[Componente] CHECK CONSTRAINT
[FK_COMPONEN_COMPONENT_MARCA]
```

```
GO
```

```
ALTER TABLE [dbo].[Componente] WITH CHECK ADD CONSTRAINT
[FK_COMPONEN_EQUIPO_CO_EQUIPO] FOREIGN KEY([IdEquipo])
REFERENCES [dbo].[Equipo] ([IdEquipo])
```

```
GO
```

```
ALTER TABLE [dbo].[Componente] CHECK CONSTRAINT
[FK_COMPONEN_EQUIPO_CO_EQUIPO]
```

```
GO
```

```
ALTER TABLE [dbo].[Conectividad] WITH CHECK ADD CONSTRAINT
[FK_CONECTIV_REFERENCE_EQUIPO] FOREIGN KEY([IdEquipo])
REFERENCES [dbo].[Equipo] ([IdEquipo])
```

```
GO
```

```
ALTER TABLE [dbo].[Conectividad] CHECK CONSTRAINT
[FK_CONECTIV_REFERENCE_EQUIPO]
```

```
GO
```

```
ALTER TABLE [dbo].[Equipo] WITH CHECK ADD CONSTRAINT
[FK_EQUIPO_REFERENCE_CAMPUS] FOREIGN KEY([IdCampus])
REFERENCES [dbo].[Campus] ([IdCampus])
```

```
GO
```

```
ALTER TABLE [dbo].[Equipo] CHECK CONSTRAINT
[FK_EQUIPO_REFERENCE_CAMPUS]
```

```
GO
```

```
ALTER TABLE [dbo].[Equipo] WITH CHECK ADD CONSTRAINT
[FK_EQUIPO_REFERENCE_DEPENDEN] FOREIGN KEY([IdDependencia])
REFERENCES [dbo].[Dependencia] ([IdDependencia])
```

```
GO
```

```
ALTER TABLE [dbo].[Equipo] CHECK CONSTRAINT
[FK_EQUIPO_REFERENCE_DEPENDEN]
```

```

GO
ALTER TABLE [dbo].[Equipo] WITH CHECK ADD CONSTRAINT
[FK_EQUIPO_REFERENCE_MARCA] FOREIGN KEY([IdMarca])
REFERENCES [dbo].[Marca] ([IdMarca])
GO
ALTER TABLE [dbo].[Equipo] CHECK CONSTRAINT
[FK_EQUIPO_REFERENCE_MARCA]
GO
ALTER TABLE [dbo].[Equipo] WITH CHECK ADD CONSTRAINT
[FK_EQUIPO_REFERENCE_MODELO] FOREIGN KEY([IdModelo])
REFERENCES [dbo].[Modelo] ([IdModelo])
GO
ALTER TABLE [dbo].[Equipo] CHECK CONSTRAINT
[FK_EQUIPO_REFERENCE_MODELO]
GO
ALTER TABLE [dbo].[Periferico] WITH CHECK ADD CONSTRAINT
[FK_PERIFERI_PERIFERIC_MARCA] FOREIGN KEY([IdMarca])
REFERENCES [dbo].[Marca] ([IdMarca])
GO
ALTER TABLE [dbo].[Periferico] CHECK CONSTRAINT
[FK_PERIFERI_PERIFERIC_MARCA]
GO
ALTER TABLE [dbo].[Periferico] WITH CHECK ADD CONSTRAINT
[FK_PERIFERI_PERIFERIC_MODELO] FOREIGN KEY([IdModelo])
REFERENCES [dbo].[Modelo] ([IdModelo])
GO
ALTER TABLE [dbo].[Periferico] CHECK CONSTRAINT
[FK_PERIFERI_PERIFERIC_MODELO]
GO
ALTER TABLE [dbo].[Periferico] WITH CHECK ADD CONSTRAINT
[FK_PERIFERI_REFERENCE_EQUIPO] FOREIGN KEY([IdEquipo])
REFERENCES [dbo].[Equipo] ([IdEquipo])
GO
ALTER TABLE [dbo].[Periferico] CHECK CONSTRAINT
[FK_PERIFERI_REFERENCE_EQUIPO]
GO
ALTER TABLE [dbo].[Persona] WITH CHECK ADD CONSTRAINT
[FK_PERSONA_REFERENCE_EQUIPO] FOREIGN KEY([IdEquipo])
REFERENCES [dbo].[Equipo] ([IdEquipo])
GO
ALTER TABLE [dbo].[Persona] CHECK CONSTRAINT
[FK_PERSONA_REFERENCE_EQUIPO]
GO
ALTER TABLE [dbo].[Seg_Login] WITH CHECK ADD CONSTRAINT
[FK_SEG_LOGI_LOGIN_ROL_SEG_ROL] FOREIGN KEY([IdRol])
REFERENCES [dbo].[Seg_Rol] ([IdRol])
GO

```

```

ALTER TABLE [dbo].[Seg_Login] CHECK CONSTRAINT
[FK_SEG_LOGI_LOGIN_ROL_SEG_ROL]
GO
ALTER TABLE [dbo].[Seg_Permiso] WITH CHECK ADD CONSTRAINT
[FK_SEG_PERM_PERMISO_R_SEG_ROL] FOREIGN KEY([IdRol])
REFERENCES [dbo].[Seg_Rol] ([IdRol])
GO
ALTER TABLE [dbo].[Seg_Permiso] CHECK CONSTRAINT
[FK_SEG_PERM_PERMISO_R_SEG_ROL]
GO
ALTER TABLE [dbo].[Seg_Rol_Funcion] WITH CHECK ADD CONSTRAINT
[FK_SEG_ROL__ROL_FUNCI_SEG_FUNC] FOREIGN KEY([IdFuncion])
REFERENCES [dbo].[Seg_Funcion] ([IdFuncion])
GO
ALTER TABLE [dbo].[Seg_Rol_Funcion] CHECK CONSTRAINT
[FK_SEG_ROL__ROL_FUNCI_SEG_FUNC]
GO
ALTER TABLE [dbo].[Seg_Rol_Funcion] WITH CHECK ADD CONSTRAINT
[FK_SEG_ROL__ROL_FUNCI_SEG_ROL] FOREIGN KEY([IdRol])
REFERENCES [dbo].[Seg_Rol] ([IdRol])
GO
ALTER TABLE [dbo].[Seg_Rol_Funcion] CHECK CONSTRAINT
[FK_SEG_ROL__ROL_FUNCI_SEG_ROL]
GO
ALTER TABLE [dbo].[SoftWare] WITH CHECK ADD CONSTRAINT
[FK_SOFTWARE_REFERENCE_CATEGORI] FOREIGN KEY([IdCategoriaSoftware])
REFERENCES [dbo].[Categoria_Software] ([IdCategoriaSoftware])
GO
ALTER TABLE [dbo].[SoftWare] CHECK CONSTRAINT
[FK_SOFTWARE_REFERENCE_CATEGORI]
GO
ALTER TABLE [dbo].[SoftWare] WITH CHECK ADD CONSTRAINT
[FK_SOFTWARE_REFERENCE_EQUIPO] FOREIGN KEY([IdEquipo])
REFERENCES [dbo].[Equipo] ([IdEquipo])
GO
ALTER TABLE [dbo].[SoftWare] CHECK CONSTRAINT
[FK_SOFTWARE_REFERENCE_EQUIPO]
GO

```


3. Como se encuentran distribuidos los equipos de cómputo en los diferentes departamentos.

Por cada docente existe un equipo asignado con todos los periféricos y conexión a internet

4. El tiempo que requiere para organizar la información como se lo distribuye.

Por ciclo académico

5. Existe una persona que sea responsable de llevar los registros de los equipos.

Si el encargado del departamento de Tics la Maná

6. ¿Cree usted que es importante dotarle al proporcionar al Departamento de Sistemas de Información un sistema para la gestión de la información de los equipos de cómputo?

Sería beneficioso para el departamento

7. De acuerdo a su criterio el sistema propiciara un aspecto positivo o negativo para la gestión de la información en el Departamento de Sistemas de Información.

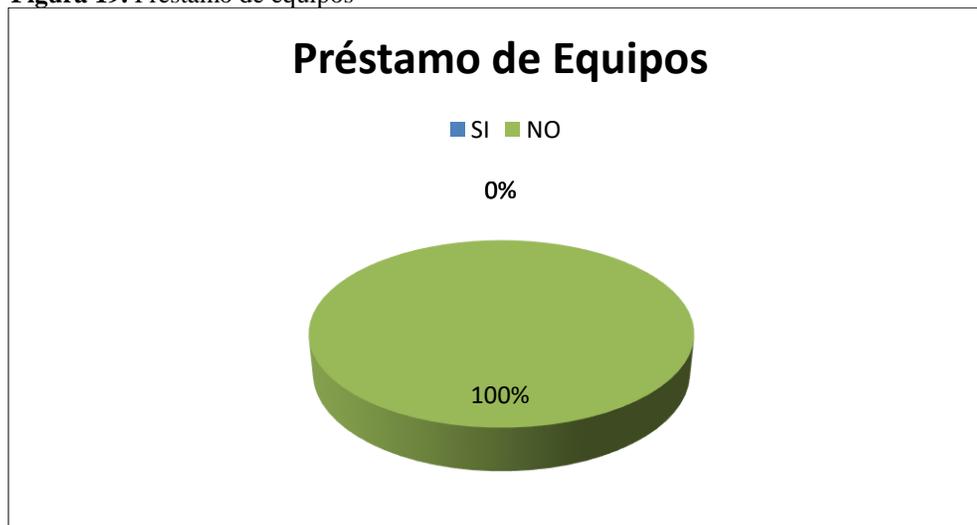
Es positivo porque ayudará a llevar un control sistematizado de los equipos

ANEXO 5.

1. **¿Tiene usted conocimiento de cómo se lleva el registro de los equipos informáticos en el Departamento de Sistemas de Información?**

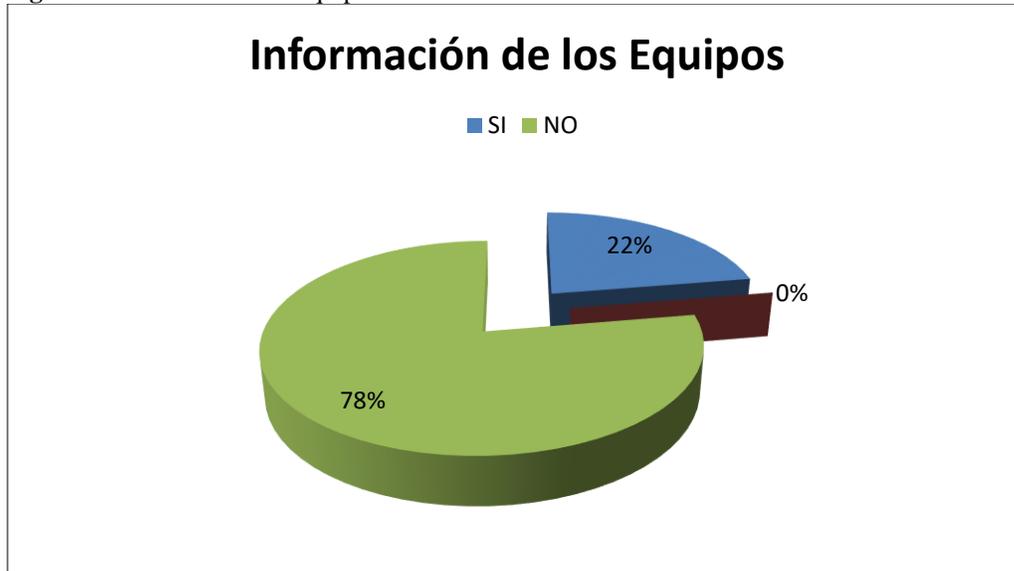
Figura 18. Registro de equipos

2. **¿Conoce usted si existen problemas al momento de registrar el préstamo de los equipos en el Departamento de Sistemas de Información?**

Figura 19. Préstamo de equipos

3. ¿Cree usted que el registro de la información de los equipos de cómputo es el adecuado?

Figura 20. Información de equipos



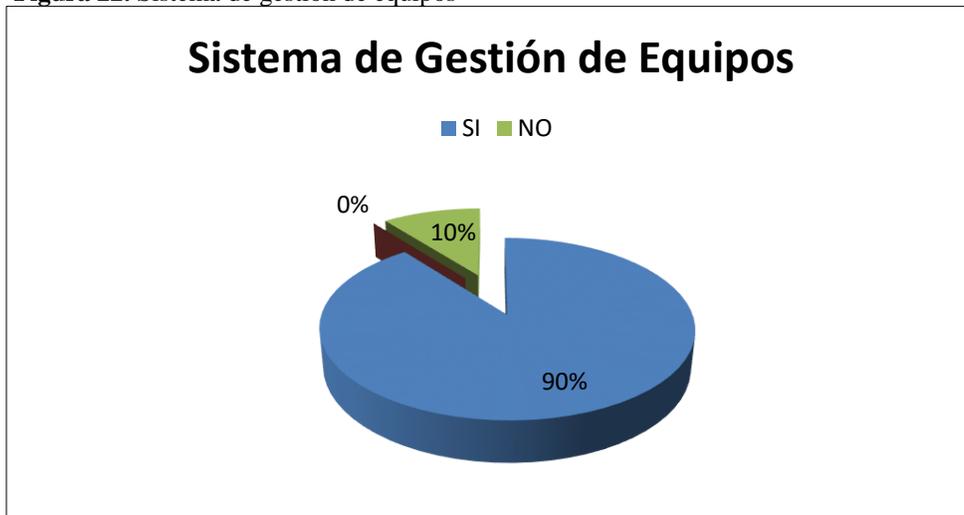
4. ¿Conoce usted sobre algún sistema el que permita llevar un registro adecuado sobres los equipos en el departamento de sistemas de información?

Figura 21. Sistema de registros de equipo



5. ¿Supone usted que sería viable la implementación de un sistema para la gestión de los equipos en el Departamento de Sistemas de Información?

Figura 22. Sistema de gestión de equipos



6. ¿Cree usted que la implementación de un sistema para la gestión de los equipos en el Departamento de Sistemas de Información permita establecer una mejor organización de la información?

Figura 23. Organización de la información



7. ¿Está usted de acuerdo que este sistema ayudará a la Institución a organizar la gestión de los equipos en el Departamento de Sistemas de Información?

Figura 24. Sistema de información



ANEXO 6. MANUAL DE USUARIO

**“DESARROLLO DE UN SISTEMA INFORMÁTICO
PARA LA GESTIÓN DE EQUIPOS TECNOLÓGICOS
DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI**

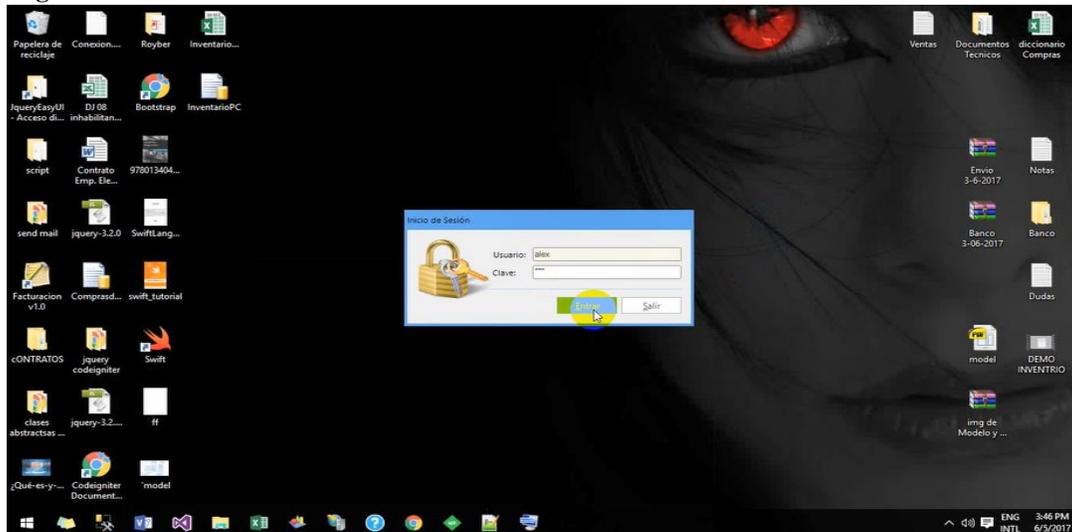
Rufino Yimael Chamaidan Panchana

2017

El Sistema informático para la gestión de equipos tecnológicos de la Universidad Técnica de Cotopaxi, posee muchas características que permiten su uso adecuado, cuyas características se describen a continuación:

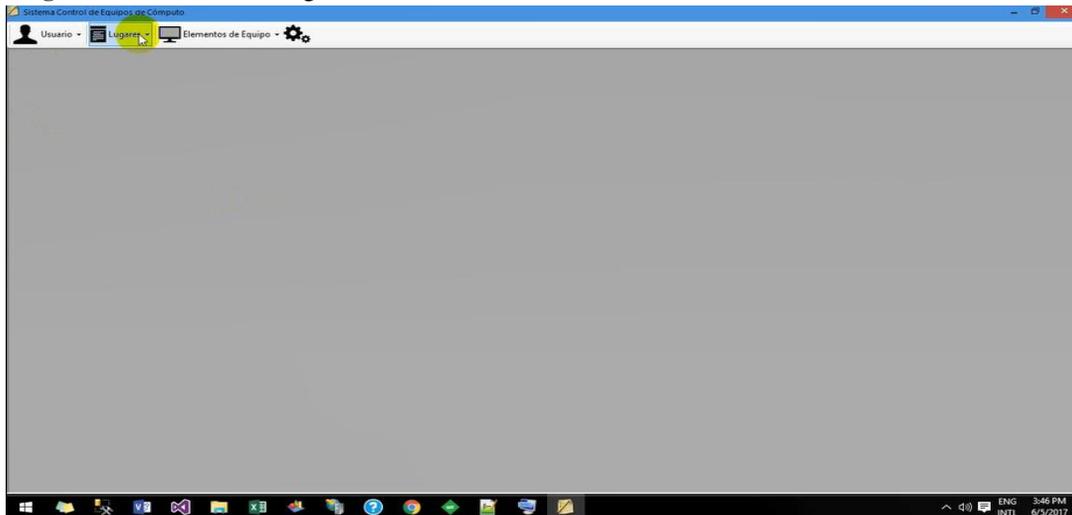
Iniciar sesión: Ingresar usuario y contraseña

Figura 25. Inicio de sesión



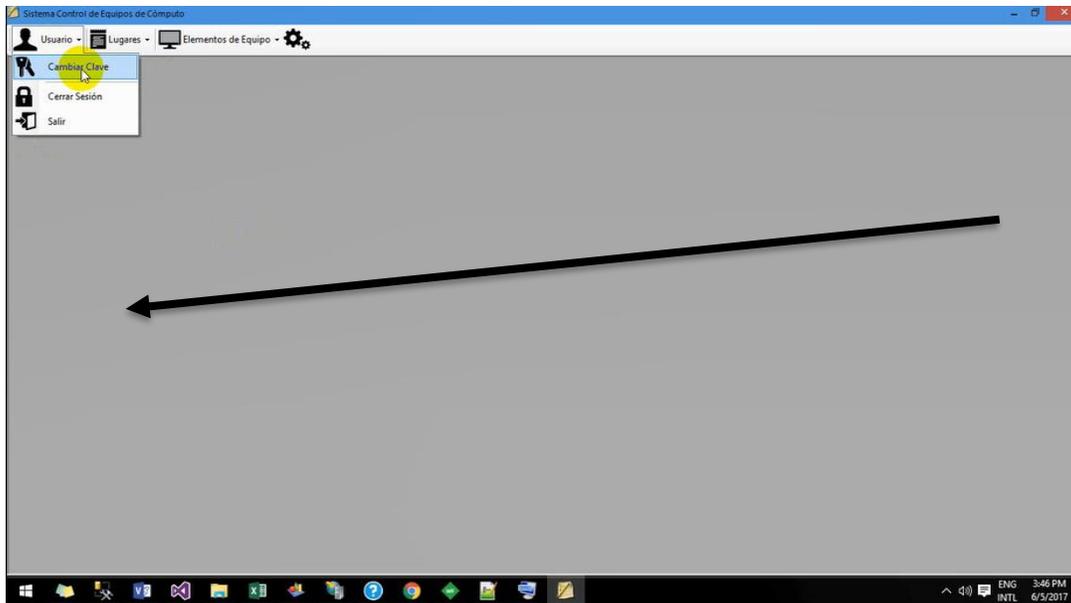
Escogemos el lugar: Seleccionar los bloques

Figura 26. Selección de lugar



Cambio de contraseña: Para el cambio de contraseña ingresar en el icono

Figura 27. Cambio de contraseña



Tipo De Componente: Seleccionar si es disco duro u otros

Figura 28. Tipo de componente

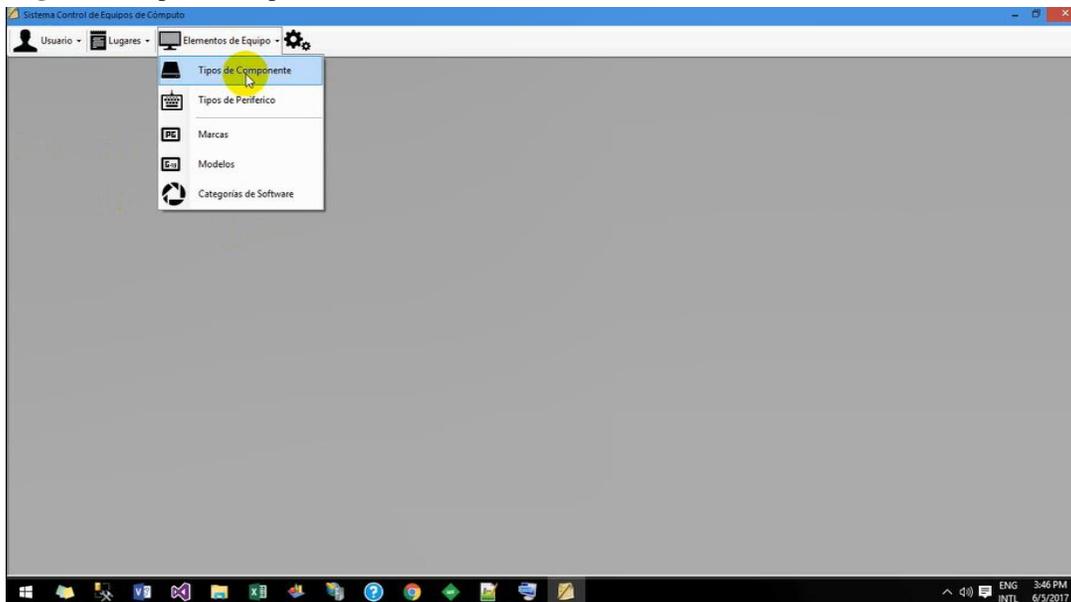


Figura 29 Selección del lugar

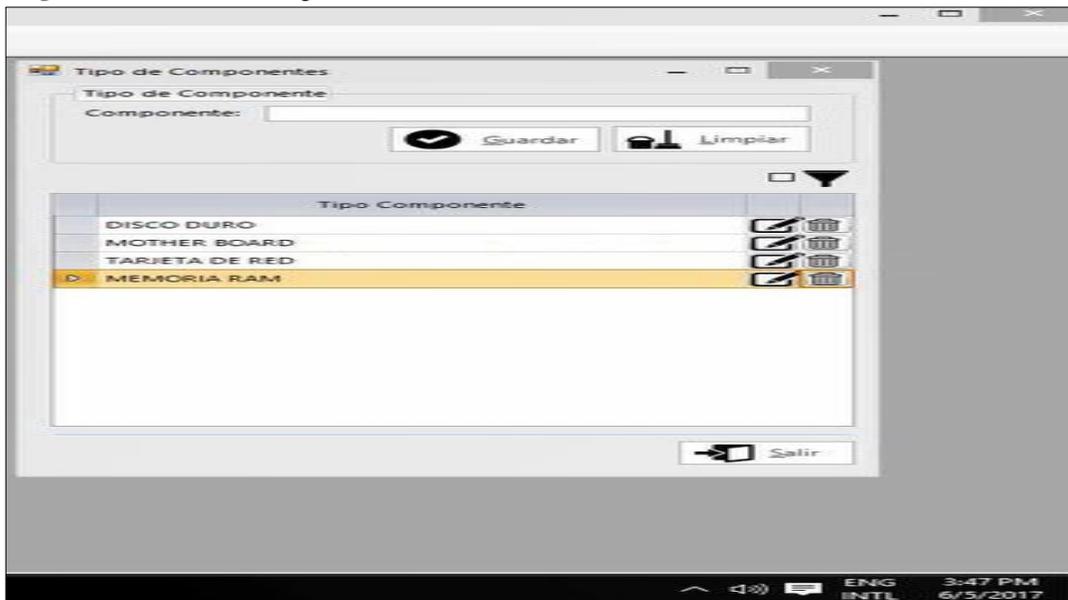
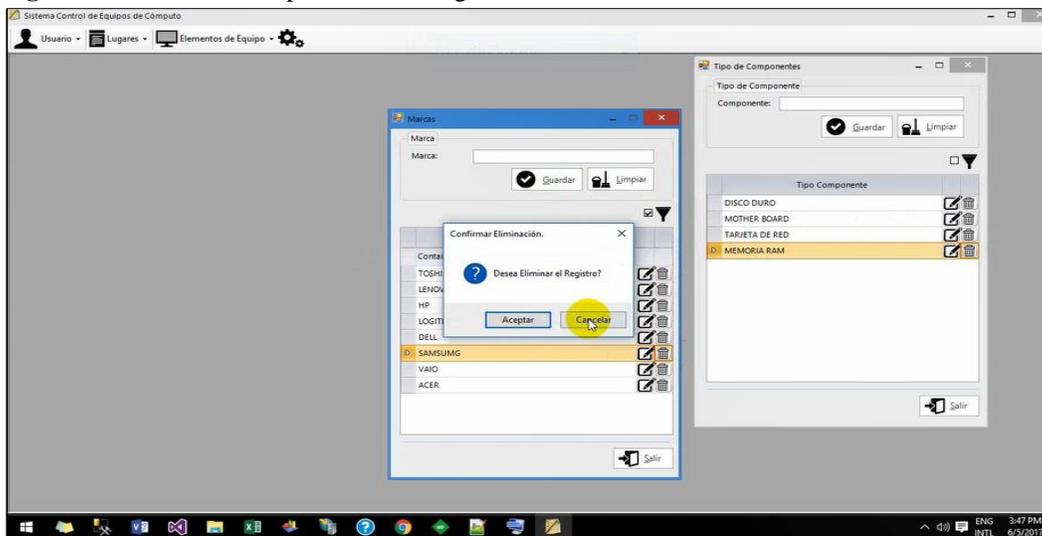
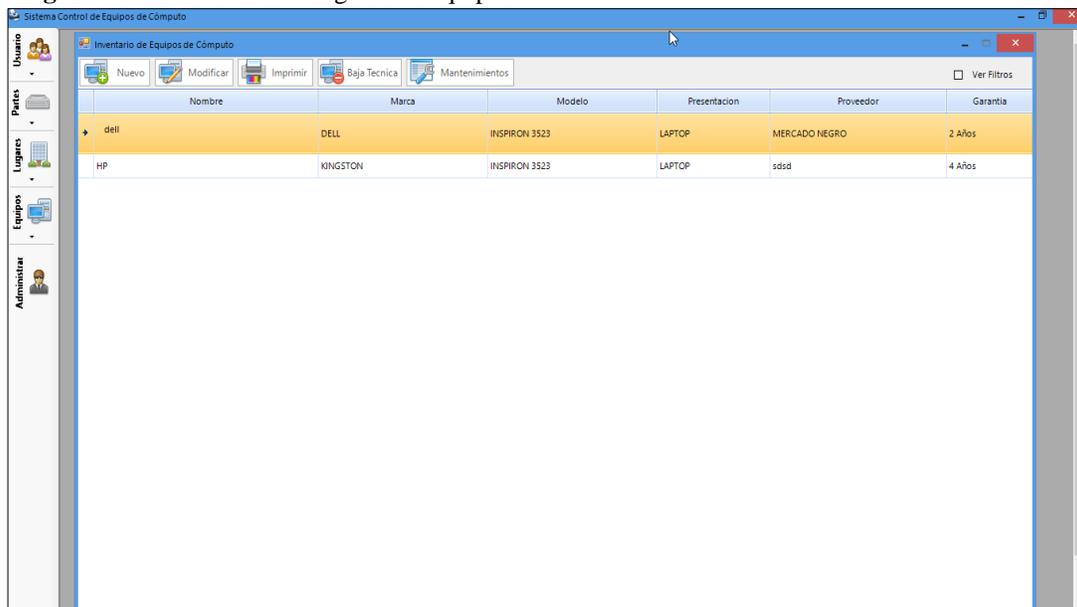


Figura 30. Procedimiento para eliminar registro



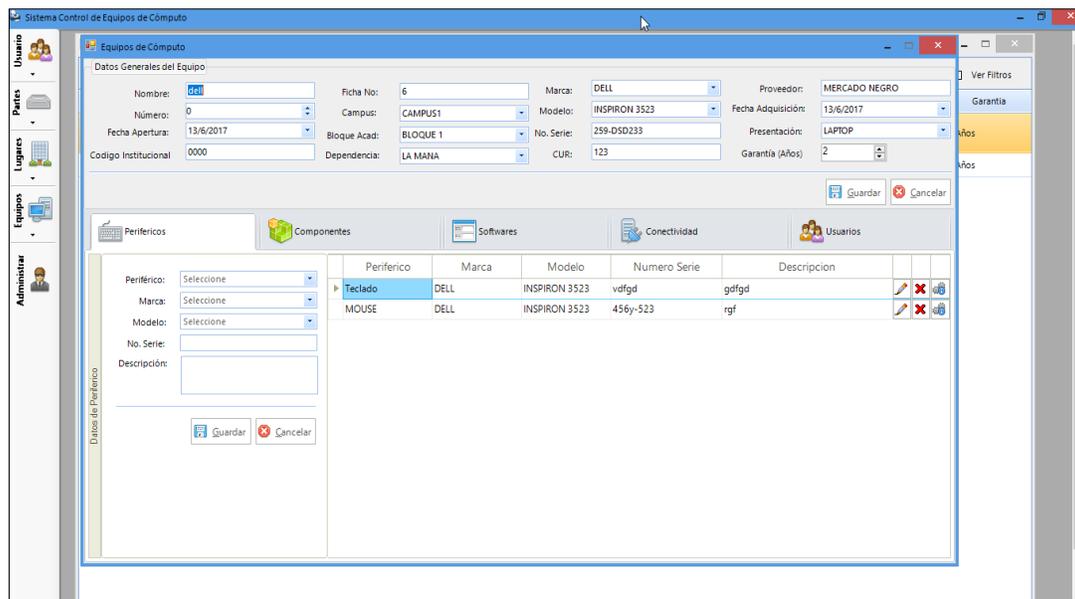
Inicio de nuevo ingreso de equipos: hacer clic en nuevo y guardar

Figura 31. Inicio de nuevo ingreso de equipos



Ingresar datos generales del equipo: Llenado de datos del equipo en registro

Figura 32. Ingreso de nuevos datos



Seleccionar datos del equipo e ingresar los ítems

Figura 33. Elección de bloque académico

Sistema Control de Equipos de Computo

Equipos de Computo

Datos Generales del Equipo

Nombre: dell Ficha No: 6 Marca: DELL Proveedor: MERCADO NEGRO
 Número: 0 Campus: CAMPUS1 Modelo: INSPIRON 3523 Fecha Adquisición: 13/6/2017
 Fecha Apertura: 13/6/2017 Bloque Acad: BLOQUE 1 No. Serie: 259-D5D233 Presentación: LAPTOP
 Código Institucional: 0000 Dependencia: LA MANA CUR: 123 Garantía (Años): 2

Perifericos

Periferico	Marca	Modelo	Numero Serie	Descripcion
Teclado	DELL	INSPIRON 3523	vdfgd	gdfgd
MOUSE	DELL	INSPIRON 3523	456y-523	rgf

Cambios de periféricos con número de series: Ingresar el código de cada equipo

Figura 34. Cambio de periférico

Sistema Control de Equipos de Computo

Equipos de Computo

Datos Generales del Equipo

Nombre: dell Ficha No: 6 Marca: DELL Proveedor: MERCADO NEGRO
 Número: 0 Campus: CAMPUS1 Modelo: INSPIRON 3523 Fecha Adquisición: 13/6/2017
 Fecha Apertura: 13/6/2017 Bloque Acad: BLOQUE 1 No. Serie: 259-D5D233 Presentación: LAPTOP
 Código Institucional: 0000 Dependencia: LA MANA CUR: 123 Garantía (Años): 2

Perifericos

Confirma que desea realizar la operación?

Se desea cambiar el periferico con numero de serie VDFGD haga click en Aceptar y seguidamente inserte el nuevo periferico.

Aceptar Cancelar

Datos de cambio de hardware: Hacer clic en el icono de cambio de hardware

Figura 35. Datos de cambio de hardware

The screenshot shows the 'Sistema Control de Equipos de Computo' interface. The main window displays 'Datos Generales del Equipo' with fields for Name, Number, Date, and various identifiers. Below this is a table of hardware components. A dialog box titled 'Cambio de Hardware' is open, allowing the user to enter a 'Fecha del Cambio' (Change Date) and a 'Descripción' (Description) for a selected component.

Periferico	Marca	Modelo	Numero Serie	Descripción
MOUSE	DELL	INSPIRON 3523	456y-523	rgf
Teclado	KINGSTON	LOG120	d1afdi	d1ahfh

Datos de mantenimiento: Para este efecto se clic en mantenimiento correctivo

Figura 36. Datos del mantenimiento

The screenshot shows the 'Sistema Control de Equipos de Computo' interface with the 'Mantenimientos' (Maintenance) section active. The 'Mto. Correctivo' (Corrective Maintenance) tab is selected. The form includes fields for 'Fecha Realizado' (Date Performed), 'Diagnostico Previo' (Previous Diagnosis), and 'Observaciones' (Observations). Below these are sections for 'Limpieza externa - interna' (External/Internal Cleaning) and 'Funcionalidad' (Functionality), each with 'Si' (Yes) and 'No' (No) radio buttons for Case, Monitor, Keyboard and Mouse, Software Verification, Hardware Verification, and Network Verification.

Historial de equipos: Aquí se puede todos los registros efectuados

Figura 37. Historial de equipos

The screenshot shows the 'Historial de Equipo' window with the following data:

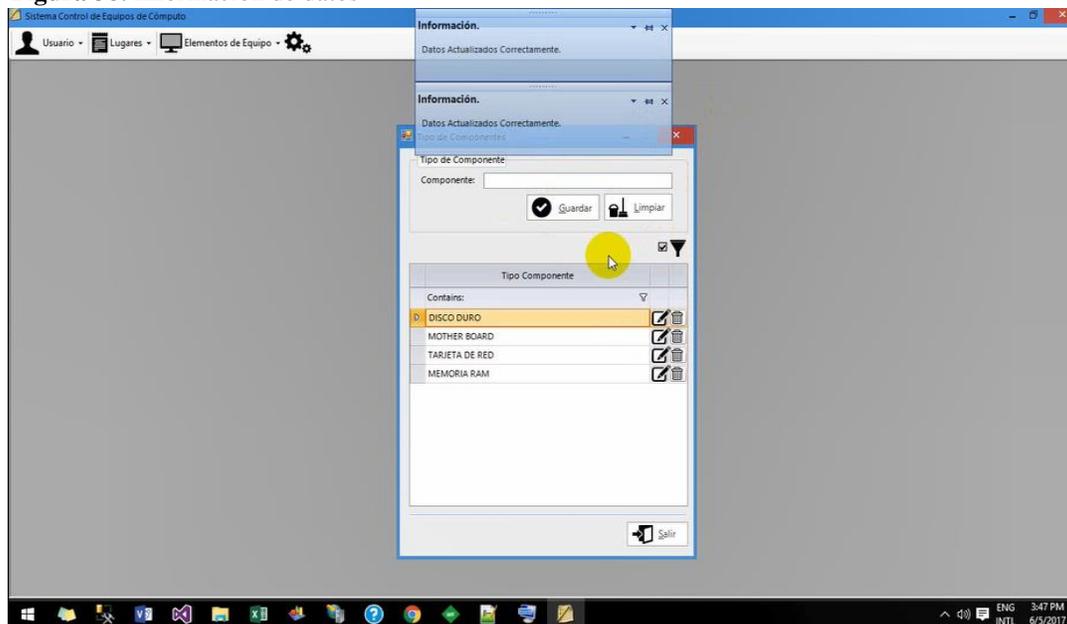
No. Equipo	Nombre	Marca	Modelo	Presentacion	Proveedor	Garantia
1	dell	DELL	INSPIRON 3523	LAPTOP	MERCADO NEGRO	2 Años
2	HP	KINGSTON	INSPIRON 3523	LAPTOP	sdsd	4 Años

Periferico	Fecha	Descripcion	Numero Serie a Reemplazar	Numero Serie Nuevo
MOUSE	23/6/2017 18:42:...		gh	gk
Teclado	23/6/2017 16:32:...	hojhdhdf	23	wrgrg
Teclado	23/6/2017 18:55:...		wrgrg	Jjgh7
MOUSE	23/6/2017 16:32:...	se cambio porque estaba roto	gk	456y-523
Teclado	23/6/2017 16:32:...	ksdndsg	vdfgd	dfdf

Periferico	Marca	Modelo	Numero Serie	Descripcion
Teclado	DELL	INSPIRON 3523	cc	fgf
MOUSE	vbn	LOG120	6ytt6	
Teclado	KINGSTON	LOG120	6666666	lkkkkkkkkkkk
Teclado	DELL	LOG120	trurtr	urjurt
Teclado	KINGSTON	LOG120	5yutr	utrur
MOUSE	xbv	LOG120	gh	ff
Teclado	KINGSTON	INSPIRON 3523	23	45

Información de datos actualizados correctamente

Figura 38. Información de datos



Datos del mantenimiento

Figura 39. Datos del mantenimiento

The screenshot displays the 'Sistema Control de Equipos de Computo' application. The main window is titled 'Inventario de Equipos de Cómputo' and contains a toolbar with 'Nuevo', 'Modificar', 'Imprimir', 'Baja Técnica', and 'Mantenimientos' buttons. Below the toolbar is a table with columns: Nombre, Marca, Modelo, Presentación, Proveedor, and Garantía. The 'Mantenimientos' section is active, showing three tabs: 'Mtto. Preventivo', 'Mtto. Correctivo', and 'Mtto. Planificado'. The 'Mtto. Preventivo' tab is selected, displaying a form with the following fields and options:

- Fecha Realizado: [dropdown menu]
- Diagnostico Previo: [text input]
- Observaciones: [text input]
- Limpeza externa - interna:
 - Case: SI NO
 - Monitor: SI NO
 - Teclado y Mouse: SI NO
- Funcionalidad:
 - Verificación de software: SI NO
 - Verificación de hardware: SI NO
 - Verificación de red: SI NO
- Buttons: Guardar, Cancelar

At the top of the form, there is a row of checkboxes: Fecha, Diagnostico Previo, Limp Case, Limp Monitor, Limp Teclado o Mouse, Verif Software, Verif Hardware, Verif Red, and Observaciones. The 'Fecha' checkbox is currently selected.