



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMA COMPUTACIONALES E

INFORMÁTICOS

Tema:

“MÓDULO DE CONTROL DE DISTRIBUTIVOS Y GENERACIÓN DE HORARIOS INTEGRADO AL SISTEMA DE CONTROL DE DOCENTES DE LA FISEI EN LA UNIVERSIDAD TÉCNICA DE AMBATO”.

Trabajo de Graduación. Modalidad: PROYECTO DE INVESTIGACIÓN.
Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

SUBLINEA DE INVESTIGACIÓN: Desarrollo de aplicaciones web

AUTOR: Sotelo Villalva Andrea Elizabeth

PROFESOR REVISOR: Ing. Mg. Franklin Oswaldo Mayorga Mayorga

Ambato -Ecuador

Agosto 2016

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de investigación sobre el tema: “**MÓDULO DE CONTROL DE DISTRIBUTIVOS Y GENERACIÓN DE HORARIOS INTEGRADO AL SISTEMA DE CONTROL DE DOCENTES DE LA FISEI EN LA UNIVERSIDAD TÉCNICA DE AMBATO**”, de la Señorita Sotelo Villalva Andrea Elizabeth, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para obtener el título terminal de tercer nivel de la Universidad Técnica de Ambato.

Ambato, Agosto del 2016

EL TUTOR

Ing. Mg. Franklin Oswaldo Mayorga Mayorga

AUTORÍA

El presente trabajo de investigación titulado: **“MÓDULO DE CONTROL DE DISTRIBUTIVOS Y GENERACIÓN DE HORARIOS INTEGRADO AL SISTEMA DE CONTROL DE DOCENTES DE LA FISEI EN LA UNIVERSIDAD TÉCNICA DE AMBATO”**. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Agosto del 2016

Andrea Elizabeth Sotelo Villalva
CC: 1803331212

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Juan Carlos Ruiz, Phd. Julio Balarezo, revisó y aprobó el informe final del trabajo de graduación titulada: **“MÓDULO DE CONTROL DE DISTRIBUTIVOS Y GENERACIÓN DE HORARIOS INTEGRADO AL SISTEMA DE CONTROL DE DOCENTES DE LA FISEI EN LA UNIVERSIDAD TÉCNICA DE AMBATO”**. Presentado por el Señorita Sotelo Villalva Andrea Elizabeth de acuerdo al Art. 17 del Reglamento de Graduación para obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

Ing. Vicente Morales, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Juan Carlos Ruiz.
DOCENTE CALIFICADOR

Phd. Julio Balarezo.
DOCENTE CALIFICADOR

DEDICATORIA

Este proyecto está dedicado a mi Padre Celestial que siempre ha estado a mi lado siendo mi fortaleza, a mis padres y mi familia, que fueron mi apoyo constante cada día brindándome lo necesario en cada etapa de mi vida.

Además a todas aquellas personas que formaron parte de mi proceso y de una u otra manera aportaron lo necesario para que pueda alcanzar un éxito más en mi vida.

Andrea Elizabeth Sotelo Villalva

AGRADECIMIENTO

Le agradezco a mi Padre Celestial por demostrarme su amor incondicional cada día y dotarme de fortaleza, perseverancia, constancia; demostrando la gran guerrera que el hizo de mí. A mis padres por haber sido mis primeros maestros enseñándome a luchar por cada sueño, anhelo y meta que me proponga y venciendo cada obstáculo para poder alcanzarlo.

A los docentes de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial por la formación profesional y personal que me brindaron durante el transcurso de mi carrera. En especial al Ing. Franklin Mayorga mi Tutor que fue una guía incondicional así como al Ing. Eduardo Chaso que me impulso a dar lo mejor de mí en lo profesional y personal.

Al Ing. Andrés Viera, Ing. Paul Bastidas y aquellas personas que estuvieron aportando su conocimiento y dedicación para que este proyecto se haya podido plasmar como un logro más en mi vida.

Y a todos quienes estuvieron constantemente a mi lado en los momentos buenos y malos, siendo mi apoyo y fuerza para continuar luchando.

Andrea Elizabeth Sotelo Villalva

ÍNDICE GENERAL

Tema	i
Aprobación del Tutor	ii
Autoría	iii
Aprobación de la comisión calificadora	iv
Dedicatoria	v
Agradecimiento	vi
Índice General	viii
Índice de Contenidos	viii
Índice de Tablas	xi
Índice de Gráficas	xiii
Resumen Ejecutivo	xiv
Summary	xv
Introducción	xvi

ÍNDICE DE CONTENIDOS

CAPÍTULO I	1
EL PROBLEMA	1
1.1 Tema	1
1.2 Planteamiento del Problema	1
1.3 Delimitación.....	3
1.4 Justificación	3
1.5 Objetivos	4
General	4
Específicos	4
CAPÍTULO II	5
MARCO TEÓRICO	5
2.1 Antecedentes Investigativos.....	5
2.2 Fundamentación Teórica	7
2.2.1 Software	7
2.2.2 Aplicaciones de Software.....	7
2.2.3 Modelos de vida de software.....	7
2.2.4 Los Procesos de Software	10
2.2.5 El modelo de referencia ISO 12207	11
2.2.6 Calidad de Software	13
2.2.7 Gestión de Calidad de Software.....	14
2.2.8 Modelo de Madurez y Capacidad Integrado (CMI).....	15
2.2.9 Búsqueda Primero en Anchura	18
2.2.10 Búsqueda en Profundidad	19
2.2.11 Algoritmos De Mejora Iterativa o Búsqueda Local	20
2.2.12 Simulated Annealing	21
2.2.13 Búsqueda Tabú.....	22
2.2.14 Hill Climbing	23
2.2.15 Conceptos Básicos Algoritmos Evolutivos.....	23
2.2.16 Algoritmos Genéticos.....	25

2.2.17 API	33
2.2.18 Distributivo	33
2.3 Propuesta Solución	33
CAPÍTULO III	34
METODOLOGÍA.....	34
3.1 Modalidad de la Investigación	34
3.2 Recolección de Información	35
3.2.1 Preguntas Entrevista Subdecanato	35
3.2.2 Preguntas Entrevista Coordinador de Carrera.....	35
3.2.3 Preguntas Entrevista Administrador de Redes	36
3.3 Procesamiento y análisis de datos	36
3.3.1 Proceso Preguntas Entrevista Subdecano.....	36
3.3.2 Proceso Preguntas Entrevista Coordinador de Carrera	38
3.3.3 Proceso Preguntas Entrevista Administrador de Redes	42
3.4 Desarrollo del proyecto	44
CAPÍTULO IV.....	45
DESARROLLO DE LA PROPUESTA.....	45
4.1 Tema	45
4.2 Datos Informativos	45
4.3 Antecedentes de la propuesta	45
4.4 Justificación	46
4.5 Análisis de Factibilidad	47
4.5.1 Factibilidad Operativa	47
4.5.2 Factibilidad Económica	47
4.5.3 Factibilidad Técnica	50
4.6 Fundamentación	51
4.7 Metodología	53
4.8 Modelo Operativo	56
4.8.1 Requerimientos del Sistema	56
4.8.2 Diagramas UML.....	57
4.8.3 Diagramas de casos de usos	58
4.8.4 Especificaciones de casos de usos	59
4.8.5 Diagrama de Secuencias	66

4.9 Selección del Framework o herramienta apropiada	72
4.9.1 Herramientas Proporcionadas	72
4.9.2 Herramientas Soportadas	72
4.9.3 Análisis de Base de Datos	73
4.9.4 Análisis de Lenguajes de Programación	74
4.9.5 Análisis de Métodos de Optimización	75
4.10 Diseño del Sistema	76
4.10.1 Diseño de la Base de Datos	76
4.10.2 Diccionario de Datos	77
4.11 Diseño de la Interfaz	102
4.12 Implementación	111
4.13 Pruebas	120
4.13.1 Pruebas de Caja Blanca	120
4.13.2 Pruebas de Caja Negra	121
4.14 Implantación	125
4.14.1 Pruebas de Aceptación	126
CAPÍTULO V	127
CONCLUSIONES Y RECOMENDACIONES	127
5.1 CONCLUSIONES	127
5.2 RECOMENDACIONES	129
BIBLIOGRAFÍA	130
ANEXOS	133

ÍNDICE DE TABLAS

Tabla 2.1 Procesos de acuerdo al estándar ISO/IEC 12207	12
Tabla 2.2 Procesos organizativos habilitadores del proyecto (ISO/IEC 12207)	13
Tabla 3.1 Recursos Humanos	47
Tabla 3.2 Recursos Tecnológicos	48
Tabla 3.3 Recursos Materiales	48
Tabla 3.4 Flujo de Pagos	48
Tabla 4.0 Caso de Uso: Ingreso al Módulo	59
Tabla 4.1 Caso de Uso: Ingreso de Información de Malla Curricular	60
Tabla 4.2 Caso de Uso: Modificación de Información de Malla Curricular	60
Tabla 4.3 Caso de Uso: Eliminación de Información de Malla Curricular	61
Tabla 4.4 Caso de Uso: Ingreso de la Información de Distributivos.....	61
Tabla 4.5 Caso de Uso: Modificación de la Información de Distributivos	62
Tabla 4.6 Caso de Uso: Eliminación de Información de Distributivo.....	62
Tabla 4.7 Caso de Uso: Ingreso de Disponibilidad de Docentes.....	63
Tabla 4.8 Caso de Uso: Modificación de Disponibilidad de Docentes	63
Tabla 4.9 Caso de Uso: Eliminación de Disponibilidad de Docentes	64
Tabla 4.10 Caso de Uso: Proceso de Horarios	64
Tabla 4.11 Caso de Uso: Consulta de Información	65
Tabla 4.12 Caso de Uso: Generar Reporte	65
Tabla 4.13 Caso de Uso: Imprimir Reporte.....	66
Tabla 4.14 Caso de Uso: Diccionario de Tablas Usuarios	78
Tabla 4.15 Caso de Uso: Diccionario de Tablas Universidad	78
Tabla 4.16 Caso de Uso: Diccionario de Tablas Facultad	78
Tabla 4.17 Caso de Uso: Diccionario de Tablas Carrera	79
Tabla 4.18 Caso de Uso: Diccionario de Tablas Profesor por Carrera.....	79
Tabla 4.19 Caso de Uso: Diccionario de Tablas Catalogo	80
Tabla 4.20 Caso de Uso: Diccionario de Tablas Contenido.....	80
Tabla 4.21 Caso de Uso: Diccionario de Tablas Cumplimiento	81
Tabla 4.22 Caso de Uso: Diccionario de Tablas Menú	82
Tabla 4.23 Caso de Uso: Diccionario de Tablas Rol	82
Tabla 4.24 Caso de Uso: Diccionario de Tablas Menú Rol	83
Tabla 4.25 Caso de Uso: Diccionario de Tablas Niveles	83
Tabla 4.26 Caso de Uso: Diccionario de Tablas Cronograma	84
Tabla 4.27 Caso de Uso: Diccionario de Tablas Subcronograma	84
Tabla 4.28 Caso de Uso: Diccionario de Tablas Detalle del Subcronograma	85
Tabla 4.29 Caso de Uso: Diccionario de Tablas Horario Temporal	86
Tabla 4.30 Diccionario de Tablas Recuperaciones	88
Tabla 4.31 Diccionario de Tablas Registro	89
Tabla 4.32 Diccionario de Tablas Creación de Horario	90
Tabla 4.33 Diccionario de Tablas Creación de Horario Nivel	91
Tabla 4.34 Diccionario de Tablas Res Horario	91
Tabla 4.35 Diccionario de Tablas Registro Temporal.....	92
Tabla 4.36 Diccionario de Tablas Registro Temporal AMW	92
Tabla 4.37 Diccionario de Tablas Huellas.....	93
Tabla 4.38 Diccionario de Tablas Tipo	93
Tabla 4.39 Diccionario de Tablas Validar Cédula	93

Tabla 4.40 Diccionario de Tablas Validar Login	94
Tabla 4.41 Diccionario de Tablas Aulas	94
Tabla 4.42 Diccionario de Tablas Jornadas	95
Tabla 4.43 Diccionario de Tablas Materias	95
Tabla 4.44 Diccionario de Tablas Niveles	95
Tabla 4.45 Diccionario de Tablas Malla	96
Tabla 4.46 Diccionario de Tablas Distributivo	96
Tabla 4.47 Diccionario de Tablas Distributivo HAC	97
Tabla 4.48 Diccionario de Tablas Disponibilidad	98
Tabla 4.49 Diccionario de Tablas Horarios	98
Tabla 4.50 Diccionario de Tablas Distributivo HAC	99
Tabla 4.51 Diccionario de Tablas Horario General	100
Tabla 4.52 Diccionario de Tablas Login Administrador	101
Tabla 4.53 Diccionario de Tablas Login Horarios Distributivo	101

ÍNDICE DE GRÁFICOS

Gráfico 2.0 Una formulación del modelo de ciclo de vida en cascada	8
Gráfico 2.1 Modelo en V	9
Gráfico 2.2 Modelo de proceso en espiral	10
Gráfico 2.3 Procesos de contexto del sistema	11
Gráfico 2.4 Procesos específicos del software en el estándar ISO/IEC 12207	14
Gráfico 2.5 Búsqueda primero en anchura sobre un árbol binario sencillo	19
Gráfico 2.6 Búsqueda primero en profundidad sobre un árbol binario	20
Gráfico 2.7 Ciclo principal de un Algoritmo Genético	29
Gráfico 2.8 Cruzamiento de único-punto	31
Gráfico 4.0 Ciclo de vida XP	54
Gráfico 4.1 Prácticas XP	56
Gráfico 4.2 Diagrama Caso de Uso de los Procesos del Módulo	58
Gráfico 4.3 Diagrama de Secuencia Ingreso al Módulo	67
Gráfico 4.4 Diagrama de Secuencia Ingreso de Malla	67
Gráfico 4.5 Diagrama de Secuencia Modificación de Malla	68
Gráfico 4.6 Diagrama de Secuencia Eliminación de Malla	68
Gráfico 4.7 Diagrama de Secuencia Ingreso de Distributivo	69
Gráfico 4.8 Diagrama de Secuencia Modificación de Distributivo	69
Gráfico 4.9 Diagrama de Secuencia Eliminación de Distributivo	70
Gráfico 4.10 Diagrama de Secuencia Ingreso de Disponibilidad Docentes	70
Gráfico 4.11 Diagrama de Secuencia Modificación de Disponibilidad Docentes	71
Gráfico 4.12 Diagrama de Secuencia Eliminación de Disponibilidad Docentes	71
Gráfico 4.13 Diagrama de Secuencia Proceso Horarios	72
Gráfico 4.14 Diseño de Interfaz Página Inicio	102
Gráfico 4.15 Diseño de Interfaz Página Ingreso	103
Gráfico 4.16 Diseño de Interfaz Página Ingreso	103
Gráfico 4.17 Diseño de Interfaz Página Principal	104
Gráfico 4.18 Diseño de Interfaz Administrar Usuarios	105
Gráfico 4.19 Diseño de Interfaz Administrar Aulas	106
Gráfico 4.20 Diseño de Interfaz Administrar Malla	107
Gráfico 4.21 Diseño de Interfaz Disponibilidad Docentes	108
Gráfico 4.22 Diseño de Interfaz Distributivos	109
Gráfico 4.23 Diseño de Interfaz Generación de Horarios	110
Gráfico 4.24 Conexión Base de Datos	111
Gráfico 4.25 Diagrama de Flujo Algoritmo Genético	112
Gráfico 4.26 Pruebas de Ingreso al Módulo	121
Gráfico 4.27 Diseño de Interfaz Ingreso Sistema	122
Gráfico 4.28 Diseño de Interfaz Administrar Usuario	123
Gráfico 4.29 Diseño de Interfaz Administrar Aulas	124

RESUMEN EJECUTIVO

El presente proyecto “MÓDULO DE CONTROL DE DISTRIBUTIVOS Y GENERACIÓN DE HORARIOS INTEGRADO AL SISTEMA DE CONTROL DE DOCENTES DE LA FISEI EN LA UNIVERSIDAD TÉCNICA DE AMBATO.”, está desarrollado con ASP.NET y la administración de la base de datos en Sql Server 2008 R2, el objetivo es que todos aquellos procesos para la elaboración de horarios puedan ser automatizados y las personas involucradas en los mismos puedan contar con un complemento, Con el Desarrollo e Implementación del Módulo permitirá agilizar procesos, seguridad en la información y disminuir tiempo.

Mediante este proyecto se logró integrar la información de Distributivos de Trabajo, Disponibilidad de Docentes información necesaria para la Generación de Horarios en la FISEI, la cual cuenta con una interfaz amigable y de fácil comprensión para los usuarios involucrados en el procesos anteriormente mencionados además que el modulo se puede adaptar para que funcione en diferentes facultades todo esto ser posible mediante roles previamente determinados.

Por medio de la investigación y el respectivo análisis de los procesos y la información involucrada para la elaboración de horarios, se consiguió prevenir datos erróneos además de controlar de manera más eficiente el ingreso y manejo de información así como la generación de los horarios por medio de herramientas Visual Studio y los complementos como JQuery, Ajax, JavaScript, ItextSharp que facilitaron el desarrollo del Módulo así como su buen funcionamiento.

SUMMARY

This project "DISTRIBUTIONAL CONTROL MODULE AND GENERATION OF TIMES INTEGRATED SYSTEM CONTROL OF TEACHERS OF FISEI at the Technical University of Ambato." Is developed with ASP.NET and management of the database in SQL Server 2008 R2, the goal is that all processes for the preparation of schedules can be automated and the people involved in them can have a complement, with the Development and Implementation Module will allow expedite processes, information security and decrease time.

This project was successfully integrated information Distributive Working Availability of Teachers information necessary for the generation Schedules in FISEI, which has a friendly interface and easy to understand for users involved in the processes mentioned above plus the module can be adapted to work in different faculties all this be possible by predetermined roles.

Through research and the respective analysis of the processes and information involved for timetabling, it was achieved prevent erroneous data in addition to controlling more efficiently income and information management and generation times through Visual Studio tools and accessories such as JQuery, Ajax, javascript, itextsharp that facilitated the development of the module and its proper functioning.

INTRODUCCIÓN

En la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, antes de cada periodo académico los coordinadores de carrera realizan los distributivos de trabajo; los cual consta con la información de materias, docentes, niveles paralelos, créditos que serán impartidos en cada carrera de la facultad. Los cual son presentados para la aprobación y de ser necesario se hacen pequeñas modificaciones. Para la elaboración de horarios se basan en la información de los distributivos previamente aprobados y la disponibilidad del docente. Este es un proceso que se lo hace de forma manual en donde la información se encuentra almacenada en diferentes medios y es utilizada para establecer horarios verificando que no hayan cruces de horas en el cual se invierte por lo menos una semana para hacerlo. Después de ya fijados los horarios se ingresa la información al Sistema de Control de Docentes de la FISEI.

La información de Distributivos y Disponibilidad de Docentes es una parte importante para la creación de horarios sin embargo la información anteriormente mencionada se maneja y almacena de forma independiente al Sistema de Control de Docentes de la FISEI existente; lo cual no ha permitido un control adecuado de dicha información. Cabe recalcar que al momento de ingresar la información de horarios previamente realizados se produce errores de tipeo, provocando que en los horarios exista información errónea e incluso cruces de horas; de modo que por estas razones no se ha permitido un manejo optimo en cuanto a realización de horarios retrasando su proceso.

Con la implementación del Módulo de Control de Distributivos y Generación de Horarios Integrado al Sistema de Control de Docentes de la FISEI en la Universidad Técnica de Ambato se proporcionará que la información de distributivos sea automatizada, y el proceso de horarios ya no se lo realice de forma manual, sino que con la utilización de algoritmos genéticos permita que se genere automáticamente los horarios y en cada generación que se lo haga sea mucho mejor que la anterior; teniendo horarios idóneos para ser elegidos.

La importancia de hacer estos procesos de forma automatizada es que tendremos información segura, concreta evitando datos erróneos y que se encuentre almacenada de manera centralizada y disponible para las personas involucradas en los mismos procesos. Con este módulo la Facultad de Ingeniería en Sistemas Electrónica e Industrial permitirá que cuyos procesos sean mucho más rápidos y eficaces beneficiándose docentes los docentes, estudiantes y en si la facultad, porque se maneja y entrega la información de manera ágil y oportuna.

CAPÍTULO I

EL PROBLEMA

1.1 Tema: Módulo de Control de Distributivos y Generación de Horarios Integrado al Sistema de Control de Docentes de la FISEI en la Universidad Técnica de Ambato.

1.2 Planteamiento del Problema

La información es la parte más importante en cada una de las empresas, y con el tiempo se han incrementado las formas de mantenerlas seguras estableciendo sistemas informáticos, normas y gestionando recursos necesarios. Además de establecer un control en la gran cantidad de información que se maneja a diario.

El actual estado de desarrollo de los sistemas de información hace que los mismos sean más complejos, integrados y relacionados. La administración efectiva de la Tecnología de la Información (TI) es un elemento crítico para la supervivencia y el éxito de las empresas, varias son las razones que producen esta criticidad, son por ejemplo, la dependencia que tienen las organizaciones de la información para su funcionamiento, el nivel de inversión que tienen en el área de TI, la potencialidad que tiene la TI para transformar las organizaciones, los riesgos y amenazas que en la actualidad tiene la información, la economía globalizada que exige un alto nivel de competitividad, entre otras [1].

Uno de los problemas más frecuentes en los centros de informática es la falta de una adecuada organización, que permita avanzar al ritmo de las exigencias de las organizaciones. A esto hay que agregar la situación que presentan los nuevos equipos en cuanto al uso de bases de datos, redes y sistemas de información. Lo anterior, combinado con la necesidad de una eficiente planeación estratégica y corporativa de las organizaciones, y con una descentralización de equipos y centralización de la información, ha provocado la complejidad de las decisiones, y las dimensiones de los problemas en cuanto a la mejor forma de organizar el área de computo, requieran aplicar técnicas modernas de control y administración [2].

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán [3].

Con todo lo anteriormente expuesto es oportuno indicar que la FISEI cuenta con un sistema de Control de Docentes Web el cual carece de un Módulo de Control de Distributivos y Generación de Horarios, lo que provoca que se manejen varios procesos de forma manual como la asignación de materias y horas de docentes, la generación de Horarios y el ingreso directo de información en la base de datos. Esto no ha permitido el control adecuado en la información de los Distributivos así como la elaboración de horarios manera más rápida y eficiente como se lo evidencia en el Anexo 2.

Estos procesos han originado que cada período académico el control de la información de los distributivos y horarios causen molestias por las constantes modificaciones previo al inicio de clases e incluso en el transcurso de los primeros días del nuevo ciclo con lo cual podría llegarse a producir un colapso en las actividades generando retrasos en las planificaciones y la pérdida de tiempo como se lo evidencia en el Anexo 3.

Además se invierte mucho tiempo en el Control necesario para la generación de horarios, cruces de horas que ocasionan la reorganización de horarios. Mientras se va realizando el ingreso de información o las modificaciones, puede producirse actualizaciones no adecuadas, lo cual genere conflicto con los datos administrativos que se manejan durante todo el período académico.

1.3 Delimitación

Área de Académica: Software

Línea de Investigación: Desarrollo de Software

Sublíneas de Investigación: Aplicaciones Web

Delimitación espacial: La presente investigación se desarrollara en la ciudad de Ambato dentro de la Facultad de Ingeniería de Sistemas, Electrónica e Industrial.

Delimitación temporal: La presente investigación se desarrollara en los seis meses posteriores a la aprobación del proyecto por parte del H. Consejo Directivo.

1.4 Justificación

Cada uno de los distintos procesos necesarios para el control de materias, horas y la generación de horarios son realizados cada periodo académico de forma manual en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Por lo mismo es importante implementar el Módulo de Control de Distributivos y Generación de Horarios en el sistema de Control de Docentes, para prevenir datos erróneos y ahorro de tiempo.

Además de controlar de manera más eficiente el ingreso y manejo de información así como la generación de los reportes necesarios. Con ello se beneficiarán los docentes, los estudiantes y la facultad en sí, porque se maneja y entrega la información de manera

ágil y oportuna. Este módulo se lo está adaptando a que sea multifacultad que permitirá el control en los procesos anteriormente mencionados.

Así los resultados del proyecto permitirán que el departamento de Administración de Redes de la FISEI y los docentes de la misma, puedan contar con un complemento que sirva de ayuda para establecer materias y horarios e incluso con las modificaciones que en el respectivo caso lo merezca cumpliendo con los puntos establecidos.

En la facultad de Ingeniería en Sistemas, Electrónica e Industrial es factible aplicar el Módulo de Distributivos y Horarios en el sistema de Control de Docentes porque se lo puede acoplar de manera efectiva, además que existen proyectos similares realizados los cuales contribuyen para la investigación, así también cuenta con la facilidad de la información y los equipos necesarios para el desarrollo e implementación de dicho proyecto.

1.5 Objetivos

General

Implementar el Módulo de Control de Distributivos y Generación de Horarios Integrado al Sistema de Control de Docentes de la FISEI en la Universidad Técnica de Ambato.

Específicos

- Analizar el funcionamiento de la base de datos del Sistema de Control de Docentes.
- Diseñar el Módulo de Control de Distributivos y Generación de Horarios en base al levantamiento de requerimientos.
- Desarrollar el Módulo de Control de Distributivos y Generación de Horarios.
- Pruebas del Módulo de Control de Distributivos y Generación de Horarios.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes Investigativos

En el desarrollo e implementación del Sistema Automatizado para el Registro y Control Docente de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial aplicando Nuevas Tecnologías se determinó que es necesario la actualización del sistema para el registro y control docente de la facultad, ya que permitirá optimizar el tiempo en la realización de las actividades que se han venido desarrollando, generando a la vez un ahorro económico y agilidad en los tramites; por ello se recomienda migrar el sistema actual a un sitio web, para su rápido acceso, e implantar una herramienta informática que resuelva problemas de seguridad, y de esta manera llevar un registro más completo de las actividades de la facultad [4].

En el desarrollo e implementación Algoritmo Para Gestión De Horarios De La Facultad De Ingeniería De La Universidad De Cuenca se determinó que la creación de horarios es un problema complejo que en forma manual puede resolverse en varios días o incluso semanas, esto debido a las restricciones que hay que controlar. Por esta razón es necesario automatizar este proceso. En la presente tesis se ha desarrollado el método heurístico SOLU para generar horarios el cual permite la configuración de los pesos que se le asignan a las preferencias de la facultad, profesores y materias, y su proceso de asignación de sesiones de clases para cada PMP (profesor, materia, paralelo) es equitativa. Además, tiene la posibilidad de generar varias soluciones de horarios para un mismo conjunto de datos de entrada y permite escoger la mejor solución mediante un criterio de evaluación basado en la función objetivo; por ello se recomienda antes de generar los horarios, se debe registrar toda la información necesaria en la base de datos

de los profesores, materias, paralelos, etc ; además que el buen desempeño del método SOLU depende de un distributivo bien estructurado, puesto que éste es indispensable para una correcta asignación de materias y paralelos [5].

Mediante el desarrollo de la Distribución Óptima de Horarios de Clases utilizando la técnica de Algoritmos Genéticos se concluye que la metodología AUMML requiere de un análisis de un grupo completo entre planificadores y programadores, y es la que se aplicó en el Generador de Horarios además reúne las suficientes características para poder diseñar y documentar agentes inteligentes, además se ha simplificado el aprendizaje debido a que se conoce el Lenguaje de Modelamiento Unificado, Es así que este generador ha permitido integrar dichas políticas y mediante algoritmos inteligentes, generar horarios mejorados (de acuerdo con las pruebas realizadas) en poco tiempo evitando exhaustivos controles manuales; por ello se recomienda capacitar a los usuarios que vayan a administrar el sistema a fin de evitar pérdidas de tiempo, debiendo realizarse una explicación detallada de ingreso, salida y consulta de datos, así como también los problemas que se pueden presentar en el proceso de generación de horarios [6].

Con el desarrollo e implementación del Sistema Inteligente de soporte en la Generación de Horarios Académicos para la Carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana se demostró que la utilización de algoritmos genéticos fue el resultado de una evolución constante durante el proyecto, así pues se evaluó el uso de redes neuronales, búsquedas, sistemas expertos hasta llegar a determinar cuál era la mejor forma de hacerlo; por ello se recomienda y se incentiva a la utilización de frameworks siempre y cuando sea posible, también implementar un número parametrizable de timeslots, de forma que la estructura general de los horarios sea dinámica [7].

2.2. Fundamentación Teórica

Las aplicaciones web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o del desarrollo del comercio electrónico. Además una aplicación web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser la gestión de clientes, contabilidad o inventario, a través de una página web [8].

2.2.1. Software

Software es el conjunto de programas, procedimientos y documentación relacionada que se asocia con un sistema, y especialidad con un sistema de computadora. En un sentido específico, software son los programas de computadora [9].

2.2.2. Aplicaciones del Software

Software basado en Web. Las páginas Web buscadas por un explorador son software que incorpora instrucciones ejecutables (por ejemplo, CGI, HTML, Perl, o Java), y datos (por ejemplo, hipertexto y una variedad de formatos de audio y visuales). En esencia, la red viene a ser una gran computadora que proporciona un recurso software casi ilimitado que puede ser accedido por cualquiera con un modem [10].

2.2.3. Modelos de ciclos de vida de software

Modelo en cascada

El modelo del ciclo de vida en cascada (del inglés waterfall) es la de llevar a cabo distintas fases de desarrollo en secuencia, comenzando cada una de ellas en el punto en que terminó la anterior.

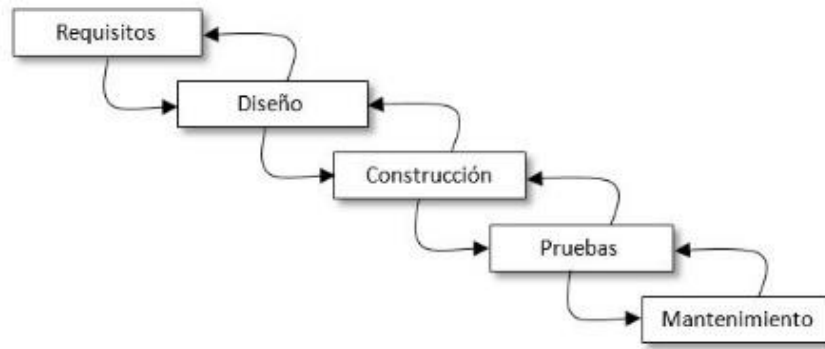


Gráfico 2.0 Una formulación del modelo de ciclo de vida en cascada
Elaborado por: Andrea Sotelo

Las ventajas del modelo en cascada incluyen las siguientes:

- El estado del proyecto se hace visible, dado que hay una progresión secuencial por fase perfectamente diferenciadas.
- Los entregables pueden asociarse a las fases, facilitando así la gestión.

Entre las desventajas tenemos las siguientes:

- Asume la estabilidad de los requisitos durante el desarrollo, lo que en la mayoría de los casos resulta poco realista.
- Los límites entre las fases son demasiado rígidos, y esto hace que la revisión de los entregables e hitos entre fases sea difícil. A menudo estas decisiones son tan complejas que obligan a paralizar el desarrollo, por ejemplo cuando se ha de dar por válido lo obtenido y proseguir, o no [9].

Modelo en V

El modelo del ciclo de vida con forma de V es una evolución del modelo en cascada en el que se enfatizan las actividades de verificación y validación.

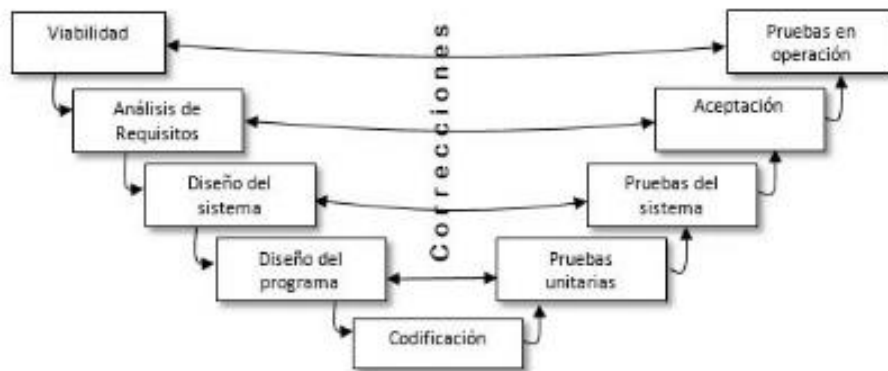


Gráfico 2.1 Modelo en V
Elaborado por: Andrea Sotelo

La parte izquierda de la V representa las fases de desarrollo previas a las actividades de prueba. En este caso, se considera que la prueba unitaria corresponde al diseño detallado o de bajo nivel, aunque en otros casos la prueba unitaria se considera como la prueba de la codificación. De hecho en las metodologías ágiles, se considera que prueba unitaria y codificaciones deberían ser actividades cuasi-simultáneas. Subiendo por la parte derecha, las pruebas del sistema serían las encargadas de evaluar el diseño de alto nivel, y las pruebas de aceptación serían la evaluación de que los requisitos que se tomaron reflejaban fielmente las necesidades de los usuarios. Por último la puesta en marcha del sistema es el contraste de que el proyecto era viable [9].

Modelos de proceso basados en prototipos

Un prototipo de software es un modelo ejecutable de un sistema futuro que implementa solo una pequeña parte de la funcionalidad, pero permite que los clientes, usuarios y desarrolladores adquirir experiencia con la arquitectura y funcionalidad [9].

Modelo en espiral

La esencia del modelo es la división en cuadrantes del gráfico, dado que cada cuadrante representa un tipo de actividad diferente:

- Identificar objetivos, alternativas y restricciones
- Evaluar las alternativas mediante la identificación y resolución de riesgos.

- Desarrollar y verificar el siguiente nivel del sistema.
- Planificar las siguientes fases [9].

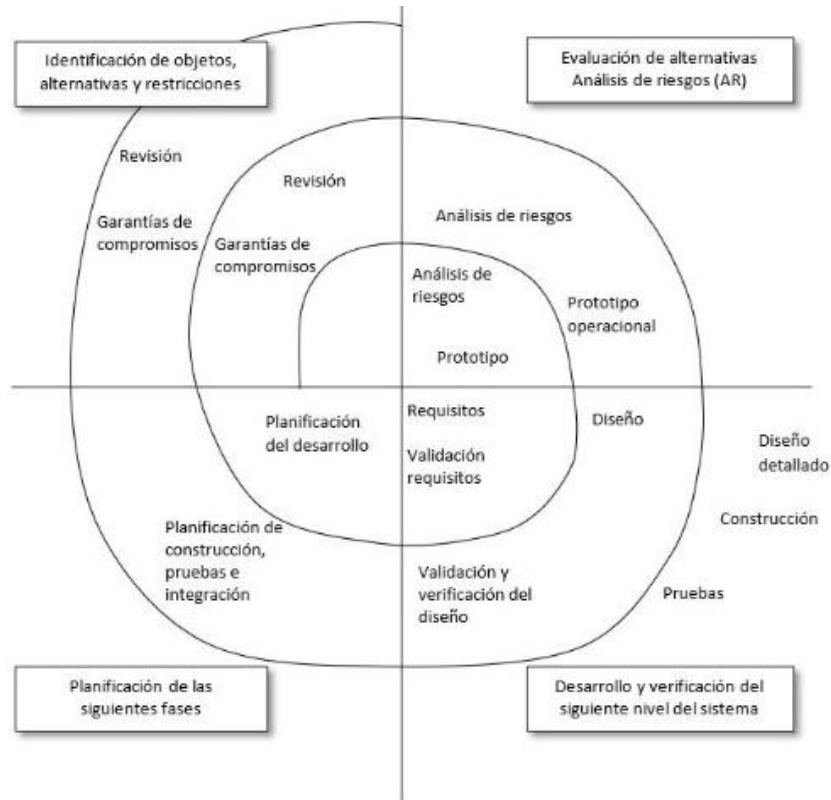


Gráfico 2.2 Modelo de proceso en espiral
Elaborado por: Andrea Sotelo

2.2.4. Los Procesos de Software

Un proceso del software es un conjunto de actividades y resultados asociados que producen un producto de software. Estas actividades son llevadas a cabo por los ingenieros de software.

Existen cuatro actividades fundamentales de procesos (incluidas más adelante en este libro) que son comunes para todos los procesos del software. Estas actividades son:

1. Especificación de software donde los clientes e ingenieros definen el software a producir y las restricciones sobre su operación.
2. Desarrollo del software donde el software se diseña y programa.

3. Validación del software donde el software se valida para asegurar que es lo que el cliente requiere.
4. Evolución del software donde el software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado [11].

2.2.5. El modelo de referencia ISO 12207

El estándar ISO/IEC 12207 es el modelo de referencia de procesos de ciclo de vida del software.

- Procesos de **contexto del sistema**: procesos de acuerdo, procesos organizativos habilitadores del proyecto, procesos de proyecto y procesos técnicos (Gráfico 4): Se trata de actividades que no son específicas de la Ingeniería del Software, pues o bien se encargan de la sistematización de las actividades que están en el contexto organizativo de ese desarrollo.

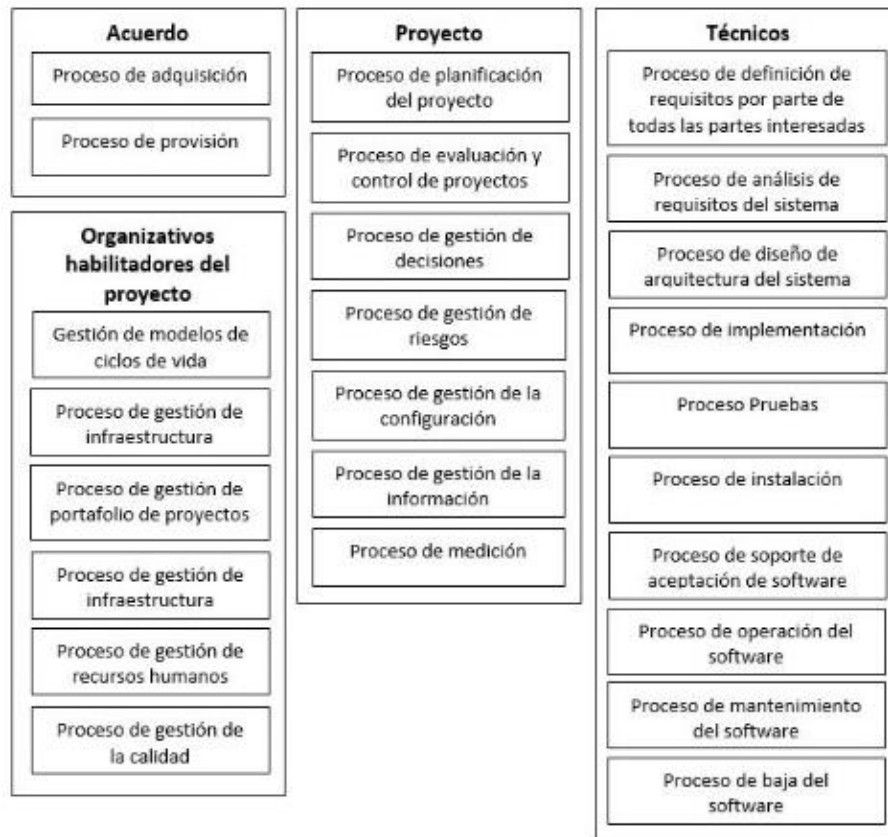


Gráfico 2.3 Procesos de contexto del sistema
Elaborado por: Andrea Sotelo

Procesos de acuerdo en el estándar ISO/IEC 12207

Proceso	Descripción
Adquisición	Es el proceso de adquisición de un bien o servicio para satisfacer unas necesidades. Este proceso implica una preparación, eventualmente analizando los requisitos y considerando diferentes opciones como la de comprar un software pre-empaquetado o hacer un desarrollo a medida. Puede implicar la selección de proveedores entre diferentes ofertas, y terminará en la negociación de un contrato. Finalmente, implica el seguimiento de lo acordado.
Provisión	Es el proceso desde la perspectiva opuesta, la del proveedor de un producto o servicio. Implica la identificación de una oportunidad, la negociación del contrato y la realización del plan y del proyecto. También implica la provisión de soporte

Tabla 2.1 Procesos de acuerdo al estándar ISO/IEC 12207

Elaborado por: Andrea Sotelo

Procesos organizativos habilitadores del proyecto (ISO/IEC 12207)

Atributo	Descripción
Gestión de modelos de ciclo de vida	Este proceso implica la definición y mantenimiento de modelos de ciclos de vida, modelos de proceso y políticas para los procesos. Las actividades fundamentales son el establecimiento del proceso, su evaluación y la mejora del mismo. Se trata de un proceso de gestión de procesos.
Proceso de gestión de infraestructura	La infraestructura incluye las instalaciones, herramientas y recursos tecnológicos necesarios en la organización para soportar la ejecución de los proyectos. Este proceso se encarga de la implantación, establecimiento y mantenimiento de la infraestructura.
Proceso de gestión del portafolio de proyectos	Este proceso tiene como objetivo la selección de un número de proyectos suficiente y adecuado para la organización. Entre sus actividades esta la identificación de oportunidades de negocio, la evaluación de la viabilidad de proyectos y su terminación bien prematura o bien según el calendario.
Proceso de gestión de recursos humanos	Incluye las actividades necesarias para que la organización cuente con el personal suficiente y competente para la realización de los proyectos, identificando las competencias necesarias, desarrollándolas cuando sea necesario, o contratando nuevo personal. También, las actividades de gestión del conocimiento y fortalecimiento de los equipos de trabajo.
Proceso de gestión de la calidad	Incluye la gestión de la calidad del proceso y del producto y la implementación de acciones correctivas

en los casos en que se necesite.

Tabla 2.2 Procesos organizativos habilitadores del proyecto (ISO/IEC 12207).

Elaborado por: Andrea Sotelo

- **Procesos específicos del software:** implementación del software, soporte y reutilización (Gráfico 5) [9].

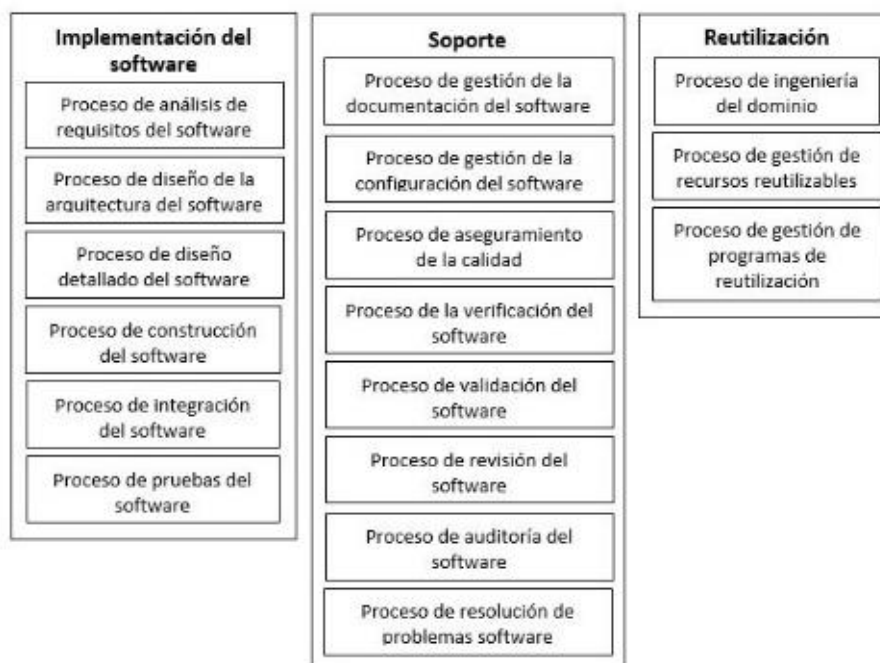


Gráfico 2.4 Procesos específicos del software en el estándar ISO/IEC 12207

Elaborado por: Andrea Sotelo

2.2.6. CALIDAD DE SOFTWARE

Pero, ¿cómo se define la calidad del software? En el sentido más general se define como: Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan.

Hay pocas dudas acerca de que la definición anterior podría modificarse o ampliarse en un debate sin fin. Para propósitos de este libro, la misma sirve a fin de enfatizar tres puntos importantes:

1. Un proceso eficaz de software establece la infraestructura que da apoyo a cualquier esfuerzo de elaboración de un producto de software de alta calidad. Los aspectos de administración del proceso generan las verificaciones y equilibrios que ayudan a

evitar que el proyecto caiga en el caos, contribuyente clave de la mala calidad. Las prácticas de ingeniería de software permiten al desarrollador analizar el problema y diseñar una solución sólida, ambas actividades críticas de la construcción de software de alta calidad. Por último, las actividades sombrija, tales como administración del cambio y revisiones técnicas, tienen tanto que ver con la calidad como cualquier otra parte de la práctica de la ingeniería de software.

2. Un producto útil entrega contenido, funciones y características que el usuario final desea; sin embargo, de igual importancia es que entrega estos activos en forma confiable y libre de errores. Un producto útil siempre satisface los requerimientos establecidos en forma explícita por los participantes. Además, satisface el conjunto de requerimientos (por ejemplo, la facilidad de uso) con los que se espera que cuente el software de alta calidad.
3. Al agregar valor para el productor y para el usuario de un producto, el software de alta calidad proporciona beneficios a la organización que lo produce y a la comunidad de usuarios finales. La organización que elabora el software obtiene valor agregado porque el software de alta calidad requiere un menor esfuerzo de mantenimiento, menos errores que corregir y poca asistencia al cliente. Esto permite que los ingenieros de software dediquen más tiempo a crear nuevas aplicaciones y menos a repetir trabajos mal hechos. La comunidad de usuarios obtiene valor agregado porque la aplicación provee una capacidad útil en forma tal que agiliza algún proceso de negocios. El resultado final es 1) mayores utilidades por el producto de software, 2) más rentabilidad cuando una aplicación apoya un proceso de negocios y 3) mejor disponibilidad de información, que es crucial para el negocio [12].

2.2.7. GESTIÓN DE CALIDAD DE SOFTWARE

Se utiliza para describir un sistema que relaciona un conjunto de variables relevantes para la puesta en práctica de una serie de principios, prácticas y técnicas para la mejora de la calidad. Así pues, el contenido de los distintos enfoques de Gestión de la Calidad se distingue por tres dimensiones

1. Los principios que asumen y que guían la acción organizativa.

2. Las prácticas –actividades– que incorporan para llevar a la práctica estos principios.
3. Las técnicas que intentan hacer efectivas estas prácticas.

Por ejemplo, un principio como la orientación hacia el cliente, asumido en diversos enfoques, puede conducir a que la organización lleve a cabo prácticas como la recogida sistemática de información sobre las necesidades, expectativas y satisfacción del cliente, que se hacen efectivas a través de estudios de mercado, pruebas de gusto en mercados seleccionados antes del lanzamiento de un nuevo [13].

2.2.8. MODELO DE MADUREZ Y CAPACIDAD INTEGRADO (CMI)

Es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento. CMMI para desarrollo es un modelo de referencia que cubre las actividades del desarrollo y del mantenimiento aplicadas tanto a los productos como a los servicios [14].

Ingeniería

Las áreas de proceso de Ingeniería cubren las actividades de desarrollo y de mantenimiento que se comparten entre las disciplinas de ingeniería.

Las áreas de proceso de Ingeniería fueron escritas usando terminología general de ingeniería de tal forma que cualquier disciplina técnica implicada en el proceso de desarrollo del producto (p.ej., ingeniería del software o ingeniería mecánica) pueda usarlas para la mejora de procesos.

Las áreas de proceso de Ingeniería también integran los procesos asociados con diferentes disciplinas de ingeniería en un único proceso de desarrollo de producto, dando soporte a una estrategia de mejora de procesos orientada a producto. Esta estrategia apunta a los objetivos de negocio esenciales más que a disciplinas técnicas específicas.

Este enfoque a procesos evita de manera eficaz la tendencia hacia una mentalidad “compartimentada” de la organización.

Las áreas de proceso de Ingeniería se aplican al desarrollo de cualquier producto o servicio dentro del dominio de desarrollo (p.ej., productos software, productos hardware, servicios o procesos).

El fundamento técnico para IPPD está basado en una solución robusta a la ingeniería de sistemas que abarca el desarrollo en el contexto de las fases de la vida del producto.

Las áreas de proceso de Ingeniería proporcionan este fundamento técnico. La implementación de IPPD se trata posteriormente a través de ampliaciones a las prácticas específicas en las áreas de proceso de Ingeniería que enfatizan el desarrollo concurrente y se focalizan en todas las fases de la vida del producto.

Las áreas de proceso de Ingeniería de CMMI son las siguientes:

- Desarrollo de requerimientos.
- Gestión de requerimientos.
- Solución técnica.
- Integración de producto.
- Verificación.
- Validación.

El área de proceso de Desarrollo de requerimientos identifica las necesidades del cliente y traduce dichas necesidades en requerimientos del producto. El conjunto de requerimientos del producto se analiza para producir una solución conceptual de alto nivel. Este conjunto de requerimientos se asigna entonces para establecer un conjunto inicial de requerimientos de componentes del producto.

El área de proceso de Gestión de requerimientos mantiene los requerimientos. Describe las actividades para obtener y controlar los cambios a los requerimientos, y asegurar que otros planes y datos relevantes se mantengan actualizados. Proporciona la trazabilidad de los requerimientos desde el cliente al producto y de éste a sus componentes.

La Gestión de requerimientos asegura que los cambios a los requerimientos se reflejan en los planes, actividades y productos de trabajo del proyecto. Este ciclo de

cambios puede afectar a todas las otras áreas de proceso de Ingeniería; así, la Gestión de requerimientos es una secuencia de eventos dinámica y a menudo recursiva.

El área de proceso de Gestión de requerimientos es fundamental para un proceso de diseño de ingeniería controlado y disciplinado.

El área de proceso de Solución técnica desarrolla los paquetes de datos técnicos relativos a los componentes del producto que serán usados por el área de proceso de Integración de producto o de Gestión de acuerdos con proveedores. Se examinan soluciones alternativas con la intención de seleccionar el diseño óptimo basado en criterios establecidos.

El área de proceso de Solución técnica se basa en las prácticas específicas en el área de proceso de Verificación para realizar la verificación del diseño y las revisiones entre pares durante el diseño y antes del ensamblaje final.

El área de proceso de Verificación asegura que los productos de trabajo seleccionados cumplen los requerimientos especificados.

El área de proceso de Verificación selecciona productos de trabajo y métodos de verificación que se usarán para verificar los productos de trabajo frente a los requerimientos especificados. La verificación es generalmente un proceso incremental, que comienza con la verificación de componentes del producto y normalmente concluye con la verificación de los productos ensamblados totalmente.

El área de proceso de Validación valida de manera incremental los productos frente a las necesidades del cliente. La validación puede realizarse en el entorno operacional o en un entorno operacional simulado. La coordinación con el cliente sobre los requerimientos de validación es un elemento importante de esta área de proceso.

El área de proceso de Integración de producto contiene las prácticas específicas asociadas con la generación de la mejor secuencia de integración posible, integrando los componentes del producto, y entregando el producto al cliente.

La Integración de producto usa las prácticas específicas de ambas, Verificación y Validación, al implementar el proceso de integración de producto. Las prácticas de verificación verifican las interfaces y los requerimientos de interfaz de los componentes

del producto antes de la integración del producto. Esto es un evento esencial en el proceso de integración. Durante la integración del producto en el entorno operacional, se usan las prácticas específicas del área de proceso de Validación [14].

Recursividad e iteración de los procesos de ingeniería

La mayoría de los estándares de proceso coinciden en que se pueden aplicar los procesos de dos maneras. Estas dos maneras se denominan recursividad e iteración. La recursividad ocurre cuando un proceso se aplica a sucesivos niveles de elementos del sistema dentro de una estructura de sistema.

Los resultados de una aplicación se usan como entrada para el siguiente nivel en la estructura del sistema. Por ejemplo, el proceso de verificación se diseña para aplicarlo al producto ensamblado completo, a los componentes principales del producto, e incluso a los componentes de componentes. La profundidad con que se puede aplicar el proceso de verificación en el producto depende enteramente del tamaño y de la complejidad del producto final [14].

2.2.9. BÚSQUEDA PRIMERO EN ANCHURA

Búsqueda primero en anchura es una estrategia sencilla en la que se expande primero el nodo raíz, a continuación se expanden todos los sucesores del nodo raíz, después sus sucesores, etc. En general, se expanden todos los nodos a una profundidad en el árbol de búsqueda antes de expandir cualquier nodo del próximo nivel.

La búsqueda primero en anchura se puede implementar llamando a la BUSQUEDA-ARBOLES con una frontera vacía que sea una cola primero en entrar primero en salir (FIFO), asegurando que los nodos primeros visitados serán los primeros expandidos. En otras palabras, llamando a la BUSQUEDA-ÁRBOLES (problema, COLA-FIFO) resulta una búsqueda primero en anchura. La cola FIFO pone a todos los nuevos sucesores generados al fin de la cola, lo que significa que los nodos más superficiales se expanden antes que los nodos más profundos [15].

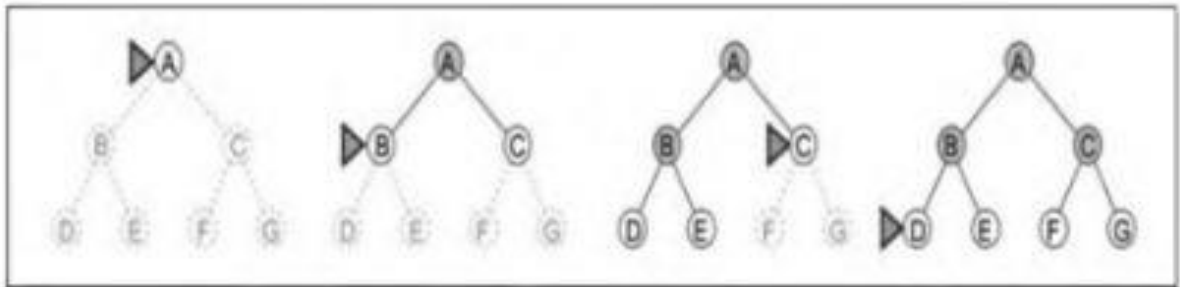


Gráfico 2.5 Búsqueda primero en anchura sobre un árbol binario sencillo.
Elaborado por: Andrea Sotelo

2.2.10. BUSQUEDA EN PROFUNDIDAD

La búsqueda primero en profundidad siempre expande el nodo más profundo en la frontera actual del árbol de búsqueda. El progreso de la se se ilustra en la Figura 3.12. La búsqueda procede inmediatamente al nivel más profundo del árbol de búsqueda, donde los nodos no tienen ningún sucesor. Cuando esos nodos se expanden, son quitados de la frontera, así entonces la búsqueda “retrocede” al siguiente nodo más superficial que todavía tenga sucesores inexplorados.

Esta estrategia puede implementarse por la BUSQUEDA-ÁRBOLES con una cola último en entrar primero en salir (LIFO), también conocida como una pila. Como una alternativa a la implementación de la BUSQUEDA-ÁRBOLES, es común aplicar la búsqueda primero en profundidad con una función recursiva que se llama en cada uno de sus hijos. (Un algoritmo primero en profundidad recursivo incorporando un límite de profundidad se muestra en Gráfico 2.7) [15].

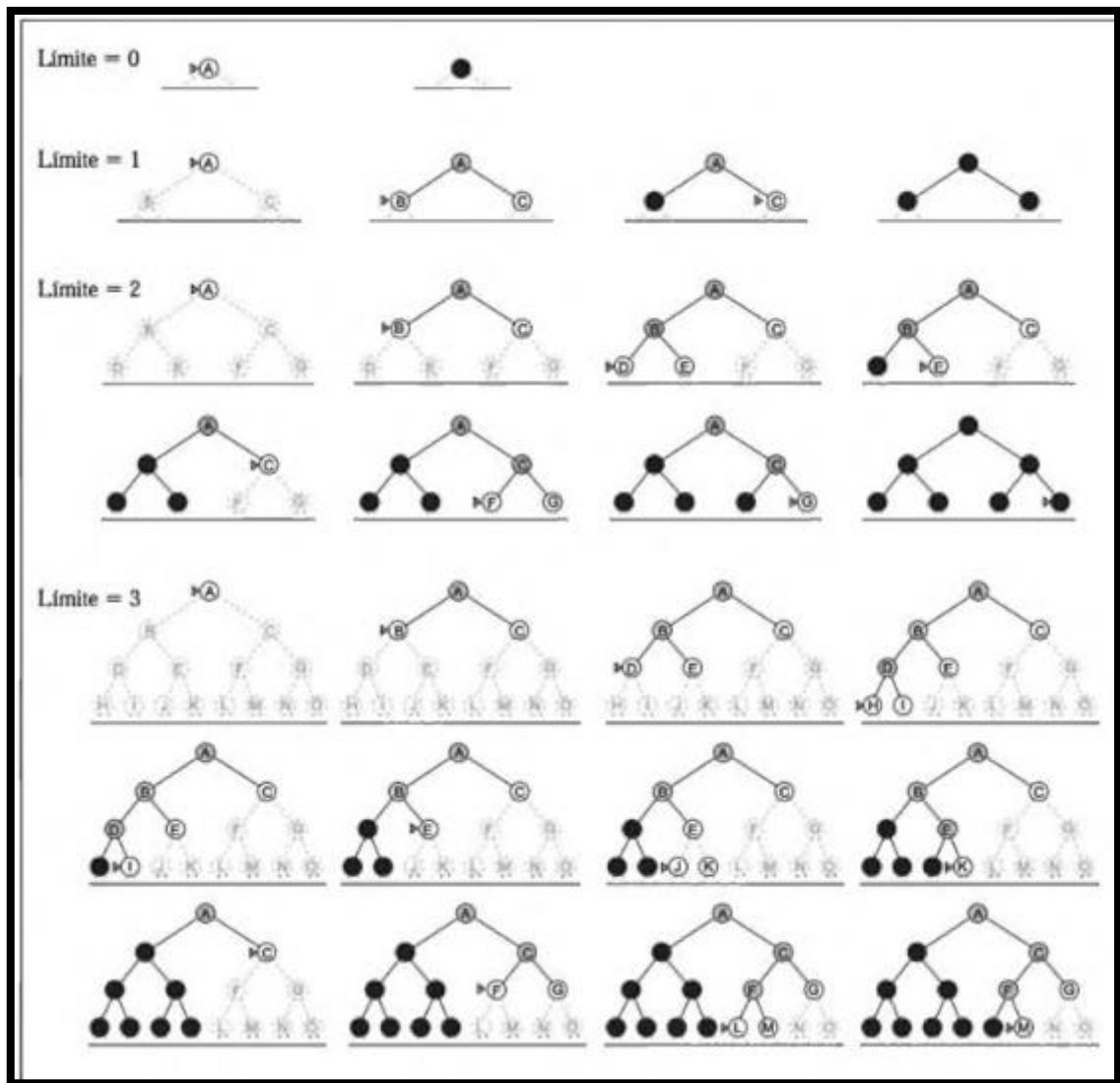


Gráfico 2.6 Búsqueda primero en profundidad sobre un árbol binario.
Elaborado por: Andrea Sotelo

2.2.11. ALGORITMOS DE MEJORA ITERATIVA O BÚSQUEDA LOCAL

En general, la solución que proporciona un algoritmo de búsqueda viene dada por un nodo objetivo junto con un camino desde el nodo inicial hasta este nodo. Sin embargo, hay muchos problemas para los cuales el nodo solución contiene toda la información necesaria y en este caso el camino es irrelevante.

La estructura básica de un método de búsqueda local es la que se muestra en el Algoritmo 9.7. El algoritmo parte de una solución inicial y en cada iteración calcula un conjunto de soluciones vecinas mediante una regla de vecindad. Cada una de estas

soluciones debe ser evaluada, siendo ésta una de las acciones más críticas del algoritmo por el elevado tiempo de ejecución que puede suponer, especialmente si el número de soluciones vecinas es muy grande, o bien si el procedimiento de evaluación de cada solución es muy costoso. A continuación, se selecciona una de las soluciones vecinas con un determinado criterio, normalmente la que tiene menor coste. Si esta solución cumple el criterio de aceptación, que puede ser simplemente que sea mejor que la solución actual S , la solución seleccionada reemplaza a la solución S , y el proceso continúa hasta que se cumpla el criterio de finalización. Este criterio suele ser que se agote un determinado número de iteraciones, o bien que no se produzcan mejoras en los últimos intentos. La regla de vecindad debe cumplir la propiedad de conectividad, es decir que desde cualquier estado se pueda alcanzar un objetivo óptimo mediante una secuencia de transformaciones. Además, las reglas suelen ser simétricas, es decir que si una solución x es vecina de otra y , y también es vecina de x [16].

Algoritmo 9.7 Esquema de un algoritmo de búsqueda local.

```

S = SolucionInicial;
mientras (no CriterioDeTerminación) hacer
    V = SolucionesVecinas(S);
    EvaluarSoluciones(V );
    S1= Seleccion(V );
    si (CriterioDeAceptación) entonces
        S = S1;
    fin si
fin mientras
devolver S;

```

2.2.12. SIMULATED ANNEALING

Está basado en el enfriamiento de los cristales (Kirkpatrick et al., 1983). El algoritmo requiere de una temperatura inicial, una final y una función de variación de la temperatura. Dicha función de variación es crucial para el buen desempeño del algoritmo y su definición es, por tanto, sumamente importante. Éste es un algoritmo probabilístico de búsqueda local.

Su principio está basado en el uso del algoritmo de Metropolis que aplicando Simulación Monte Carlo, calcula el cambio de energía que ocurre durante el enfriamiento de un sistema físico.

Algoritmo de Metrópolis: Se genera una perturbación y se calcula el cambio de energía, si ésta decrece, el nuevo estado es aceptado, caso contrario, la aceptación estará sujeta a un sorteo con la probabilidad de la Ecuación:

$$P(\delta E) = e^{-\frac{\delta E}{kt}}$$

El algoritmo representa el proceso de Simulated Annealing [17]

```

Entrada: f : la funcion objetivo a minimizar
Dato: kmax : numero maximo de iteraciones
Dato: emax : energia maxima
Dato: xnew el nuevo elemento creado
Dato: x0 e l mejor elemento conocido
Salida : x e l mejor elemento encontrado
x$ \leftarrow x0 , e$ \leftarrow E ( x )
xbest $ \leftarrow x , ebest$ \leftarrow e
k = 0
mientras k < kmax y e > emax hacer
    encontrar xnew = vecino ( x )
    enew $ \leftarrow E ( xnew )
    si P( e , enew , T( k/kmax) ) > U( t )
        entonces
            x$ \leftarrow xnew
            e$ \leftarrow enew
        fin si
    si enew < ebest entonces
        xbest $ \leftarrow xnew , ebest $
            \leftarrow enew
    fin si
    k = k+1
fin mientras [17]

```

2.2.13. BÚSQUEDA TABÚ

La Búsqueda Tabú (Tabu Search) (Glover and Laguna, 1998) en realidad es una meta-heurística, porque es un procedimiento que debe acoplarse a otra técnica, ya que no funciona por sí sola. La búsqueda tabú usa una memoria para guiar la búsqueda, de tal forma que algunas soluciones examinadas recientemente son memorizadas y tomadas como tabú (prohibidas) a la hora de tomar decisiones acerca del siguiente punto de

búsqueda. La búsqueda tabú es determinista, aunque se le pueden agregar elementos probabilísticos [17].

2.2.14. HILL CLIMBING

El algoritmo de Escalando la Colina o Hill Climbing (Russell and Norvig, 2002) es una técnica simple de búsqueda local y optimización aplicado para una función f de un único objetivo en un espacio de problema X . Haciendo iteraciones a partir de un punto inicial $x = x_0$, que usualmente es la mejor solución conocida para el problema, se obtiene un nuevo descendiente x_{new} en la vecindad de x . Si el individuo nuevo x_{new} es mejor que el predecesor x , lo reemplaza y así sucesivamente. Este algoritmo no tiene retroceso ni lleva ningún tipo de registro histórico (aunque estos y otros aditamentos son susceptibles de ser incorporados). Es importante resaltar que Hill Climbing busca maximizar o minimizar la función $f(x)$ donde x es discretizado, o sea $x \in X \subseteq N_p$.

Se puede decir que Hill Climbing es una forma simple de búsqueda de la dirección de gradiente, por esto fácilmente puede quedar atrapado en óptimos locales [17].

El algoritmo Hill Climbing (f)

```
Requiere:  $f$  : la función objetivo a minimizar
Requiere:  $n$  : número de iteraciones
Dato:  $x_{new}$  el nuevo elemento creado
Dato:  $x_0$  el mejor elemento conocido
Salida:  $x$  el mejor elemento encontrado
 $i = 0$ 
repetir
     $x = x_0$ 
    Evaluar  $f(x)$ 
    Encontrar un  $x_{new} = \text{vecino}(x)$ 
    Evaluar  $x_{new}$ 
    si  $f(x_{new}) > f(x)$  entonces
         $x = x_{new}$ 
    fin si
 $i = i + 1$ 
hasta que  $i \geq n$  [17].
```

2.2.15. CONCEPTOS BÁSICOS ALGORITMOS EVOLUTIVOS

Genotipo: expresión genética de un organismo o estructura genética del organismo. La información contenida en el genoma.

Fenotipo: características físicas de un organismo, atribuibles a la expresión de su fenotipo. Contiene tanto los rasgos físicos como los conductuales. Es el resultado de la

interacción entre el genotipo y el ambiente; se interpreta como la suma de los caracteres observables en un individuo. Es la manifestación externa del genotipo.

Cromosoma: es la molécula única de ADN, unida a histonas (proteínas básicas) y otras proteínas que se condensa durante la mitosis (proceso de división celular- reparto equitativo del material hereditario) y la meiosis (proceso de fragmentación —divisiones pequeñas), formando una estructura compacta.

Gen: especifica la herencia de un carácter; está formado por una secuencia de aminoácidos de una o más cadenas de ARN (ácido ribonucleico: interviene en diferentes neuronas, en la expresión de la información genética), que realizan diferentes funciones en la cadena.

Alelo: el valor de un gen. Una de las dos o más formas alternativas de un gen; determina el carácter controlado por el gen. Un ejemplo es el diploide que contiene dos juegos de cromosomas, por lo tanto tiene dos copias de cada gen.

Función de aptitud: es un tipo especial de función que cuantifica la optimalidad de una solución. Se traduce en un cromosoma óptimo para que sus bases sean combinadas con cualquier otra técnica para la producción de una nueva generación que sea mejor a las anteriores. En términos generales, las cadenas de los AG son análogas a los cromosomas en el sistema biológico. En los organismos naturales, uno o más cromosomas se combinan para formar la prescripción genética total, para la construcción y operación del organismo. En los organismos naturales el “paquete total” de genética se denomina genotipo. En los AG el “paquete total” de las cadenas se llama estructura (la estructura está compuesta por varias cadenas). En los organismos naturales, la creación de los organismos se realiza mediante la interacción de “paquetes” genéticos con su medio y se llama fenotipo.

En la terminología natural, se dice que los cromosomas están compuestos por genes, los cuales permiten tomar distintos valores llamados alelos. En la genética la posición del gen (llamado locus) se identifica en forma separada de la función del gen. Así, se puede hablar de un gene en particular, por ejemplo el gen que da color a los ojos de los animales: su lugar es la posición 10 y el valor del alelo es “ojos azules” [18].

2.2.16. ALGORITMOS GENÉTICOS

Antecedentes

Los algoritmos genéticos denominados originalmente planes reproductivos genéticos fueron desarrollados por John H. Holland a principios de la década de, quien interesado en estudiar los procesos lógicos que se daban en la adaptación, e inspirado por los estudios realizados en esa época con autómatas celulares y redes neuronales, se percató que el uso de reglas simples podía conllevar a comportamientos flexibles visualizando así, la posibilidad de estudiar la evolución en sistemas complejos.

Holland se percató que, para estudiar la adaptación era necesario considerar los siguientes principios:

- a) la adaptación ocurre en un ambiente,
- b) la adaptación es un proceso poblacional,
- c) los comportamientos individuales se pueden representar como programas,
- d) se pueden generar comportamientos futuros haciendo variaciones aleatorias en los programas
- e) las salidas de los programas tienen relación entre ellas si sus respectivas estructuras también tienen relación entre sí.

Así, Holland logró ver el proceso de adaptación como un mecanismo en el que los programas de una población interactúan y van mejorando de acuerdo a un ambiente que determina su calidad. La combinación de variaciones aleatorias con un proceso de selección basado en la calidad de los programas para el ambiente, debía conducir a un sistema adaptativo general. Es este sistema lo que Holland denominó Plan Reproductivo Genético. Aunque los Algoritmos Genéticos fueron concebidos en el contexto de adaptación, parte de Machine Learning, actualmente son masivamente utilizados como herramienta de optimización [19].

En los últimos años, la comunidad científica internacional ha mostrado un creciente interés en una nueva técnica de búsqueda basada en la teoría de la evolución y que se conoce como el algoritmo genético. Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se

producen en su entorno. Hoy en día se sabe que estos cambios se efectúan en los genes de un individuo (unidad básica de codificación de cada uno de los atributos de un ser vivo), y que sus atributos más deseables (i.e., los que le permiten adaptarse mejor a su entorno) se transmiten a sus descendientes cuando éste se reproduce sexualmente [17].

Que es un algoritmo genético

Expuesto concisamente, un algoritmo genético (o AG para abreviar) es una técnica de programación que imita a la evolución biológica como estrategia para resolver problemas. Dado un problema específico a resolver, la entrada del AG es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y una métrica llamada función de aptitud que permite evaluar cuantitativamente a cada candidata. Estas candidatas pueden ser soluciones que ya se sabe que funcionan, con el objetivo de que el AG las mejore, pero se suelen generar aleatoriamente.

Luego el AG evalúa cada candidata de acuerdo con la función de aptitud. En un acervo de candidatas generadas aleatoriamente, por supuesto, la mayoría no funcionarán en absoluto, y serán eliminadas. Sin embargo, por puro azar, unas pocas pueden ser prometedoras -pueden mostrar actividad, aunque sólo sea actividad débil e imperfecta, hacia la solución del problema.

Estas candidatas prometedoras se conservan y se les permite reproducirse. Se realizan múltiples copias de ellas, pero las copias no son perfectas; se introducen cambios aleatorios durante el proceso de copia. Luego, esta descendencia digital prosigue con la siguiente generación, formando un nuevo acervo de soluciones candidatas, y son sometidas a una ronda de evaluación de aptitud. Las candidatas que han empeorado o no han mejorado con los cambios en su código son eliminadas de nuevo; pero, de nuevo, por puro azar, las variaciones aleatorias introducidas en la población pueden haber mejorado a algunos individuos, convirtiéndolos en mejores soluciones del problema, más completas o más eficientes. De nuevo, se seleccionan y copian estos individuos vencedores hacia la siguiente generación con cambios aleatorios, y el proceso se repite. Las expectativas son que la aptitud media de la población se incrementará en cada ronda

y, por tanto, repitiendo este proceso cientos o miles de rondas, pueden descubrirse soluciones muy buenas del problema.

Aunque a algunos les puede parecer asombroso y antiintuitivo, los algoritmos genéticos han demostrado ser una estrategia enormemente poderosa y exitosa para resolver problemas, demostrando de manera espectacular el poder de los principios evolutivos. Se han utilizado algoritmos genéticos en una amplia variedad de campos para desarrollar soluciones a problemas tan difíciles o más difíciles que los abordados por los diseñadores humanos. Además, las soluciones que consiguen son a menudo más eficientes, más elegantes o más complejas que nada que un ingeniero humano produciría. ¡En algunos casos, los algoritmos genéticos han producido soluciones que dejan perplejos a los programadores que escribieron los algoritmos en primera instancia [17].

Componentes de un Algoritmo Genético

[Michalewicz1996] afirma que para poder aplicar el algoritmo genético se requiere de los siguientes 5 componentes básicos:

1. Una representación de las soluciones potenciales del problema, de naturaleza genotípica, denominada cromosoma.
2. Una manera de crear una población inicial de posibles soluciones, denominada inicialización, que por lo general se basa en un proceso aleatorio.
3. Una función objetivo que hace el papel del ambiente, calificando las soluciones de acuerdo a su aptitud, denominada función de evaluación.
4. Operadores genéticos que alteren la composición de los genomas seleccionados produciendo descendientes para las siguientes generaciones.
5. Parametrizaciones, es decir valores para los diferentes parámetros que utiliza el algoritmo genético (tamaño de la población, probabilidad de aplicar crossover, mutación, número máximo de generaciones, etc.) [20].

Procedimiento Elemental

El pseudocódigo para un algoritmo básico es el siguiente:

Procedimiento de un Algoritmo Genético

$t = 0$ inicializa población de genotipos, G_t decodificar y evaluar las estructuras de G_t
 $c_{fin} = \text{falso}$, $t = t + 1$ seleccionar G_t de G_{t-1} operar G_t , formando G_t evaluar G_t

Donde

G_t es la población de cromosomas en la generación t .

t es el iterador de generaciones.

G_t es el conjunto de individuos seleccionados para reproducir (que sufrirán cruce o mutación).

c_{fin} es una condición de finalización del algoritmo

Para realizar la evaluación de un genotipo $g_i \in G_t$ es necesario decodificarlo a su fenotipo $Y_i : g_i \rightarrow X_i$ y así poder aplicar la función de evaluación $f(X_i)$.

Al cumplirse la condición de finalización el algoritmo debe dejar de iterar, y el individuo con mejor aptitud X_{best} encontrado hasta ese momento será considerado la solución obtenida por el algoritmo genético. Este criterio de finalización suele ser un número máximo de generaciones It_{max} o el cumplimiento de una prueba de convergencia aplicada a la población X_t [17].

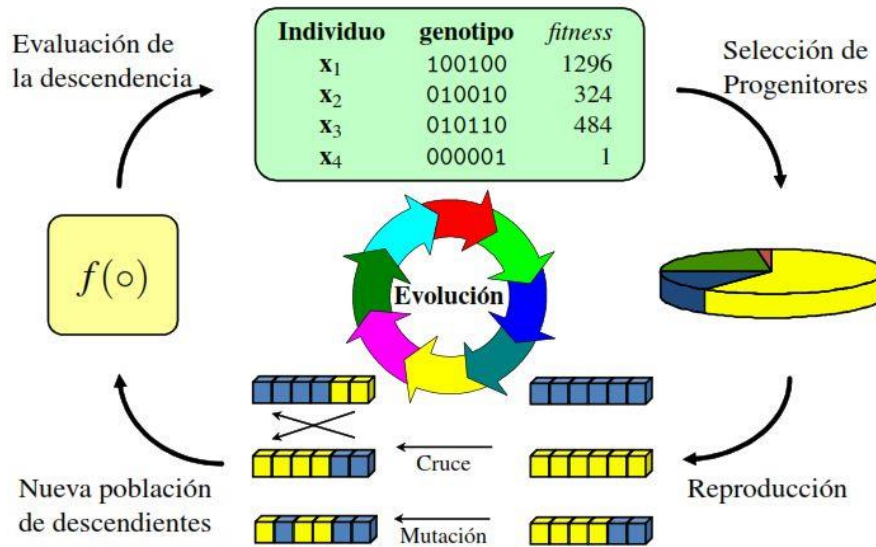


Gráfico 2.7 Ciclo principal de un Algoritmo Genético.
Elaborado por: Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn)

Selección natural

Mientras reflexionaba acerca de la selección artificial, Darwin leyó un ensayo del economista Thomas Malthus. El ensayo sugería que de no controlarse la población humana, sobrepasaría con el tiempo a la producción de alimentos, ocasionando una fuerte competencia por la existencia. Darwin se dio cuenta de que las ideas de Malthus podían aplicarse al mundo natural. Razonó que algunos competidores en la lucha por la existencia estarían mejor equipados para sobrevivir que otros. Los menos equipados morirían.

Allí estaba, finalmente, el marco de trabajo para una nueva teoría sobre el origen de las especies. La teoría de Darwin tiene cuatro principios básicos que explican cómo pueden cambiar los rasgos de una población con el tiempo. Primero, los individuos de una población muestran diferencias o variaciones. Segundo, las variaciones pueden heredarse, lo cual significa que pasan de padre a hijo. Tercero, los organismos tienen más descendientes de los que pueden sobrevivir con los recursos disponibles.

El cardenal promedio, por ejemplo, pone nueve huevos cada verano. Si cada pichón de cardenal sobrevive y se reproduce una sola vez, tomaría sólo siete años para que el primer par produjera un millón de aves.

Finalmente, las variaciones que aumentan el éxito reproductivo tendrán mayor oportunidad de transmitirse, que aquellas que no lo aumentan. Si tener una cola de abanico ayuda a las palomas a reproducirse exitosamente, futuras generaciones incluirán más palomas con este tipo de cola.

Darwin llamó a su teoría selección natural. Razonó que, dado el tiempo necesario, la selección natural podría modificar una población lo suficiente como para producir nuevas especies.

Para las estructuras básicas de un algoritmo genético también es necesario conocer la transición de una generación a otra, la cual consta de cuatro elementos básicos: Selección, Cruzamiento, Mutuación y Reemplazo [18].

1. Selección

La selección es la componente que guía el algoritmo para encontrar la solución, prefiriendo dentro de un grupo de baja función de aptitud a los más altos. Puede ocuparse una operación determinista; en la mayoría de las implementaciones tiene componentes aleatorios. La probabilidad de escoger el individuo adecuado es directamente proporcional a su función de aptitud. Se puede observar como un experimento aleatorio con

$$P[b_{j,t} \text{ es seleccionada}] = \frac{f(b_{j,t})}{\sum_{k=1} f(b_{k,t})}$$

Sólo funciona para los valores positivos de las funciones de aptitudes. Si éste no es el caso, se debe aplicar (un cambio en el caso más simple) una transformación de no-decremento $\varphi: \mathfrak{R} \rightarrow \mathfrak{R}$. Entonces la probabilidad se puede expresar por

$$P[b_{j,t} \text{ es seleccionada}] = \frac{\varphi(f(b_{j,t}))}{\sum_{k=1} \varphi(f(b_{k,t}))}$$

Las diferentes salidas se obtendrán con diferentes probabilidades. La programación del algoritmo se puede proponer como sigue, y la configuración analógica se representa con la expresión matemática anterior [18].

2. Cruzamiento

En la reproducción sexual como se desempeña en el mundo real, la materia genética de los progenitores se mezcla cuando los gametos de los progenitores se fusionan. Por lo general los cromosomas son aleatoriamente divididos y fusionados, con la consecuencia de que algunos genes de los descendientes provienen de un progenitor, mientras que otros provienen del otro progenitor. Este mecanismo se llama cruzamiento. Es una herramienta muy potente para introducir nuevos materiales genéticos y mantener la diversidad genética, pero con la notable propiedad de que progenitores saludables también producen buen rendimiento en los descendientes, o incluso mejores. Varias investigaciones han llegado a la conclusión que el cruzamiento es la razón por la que las especies de reproducción sexual se adaptan más rápido que las de reproducción asexual. Básicamente, el cruzamiento es el intercambio de genes entre los cromosomas de los dos progenitores. En un caso sencillo, se puede realizar este proceso cortando dos cadenas en una posición elegida al azar e intercambiarlas en sus extremos. Este proceso, que se nombra cruzamiento de único-punto.

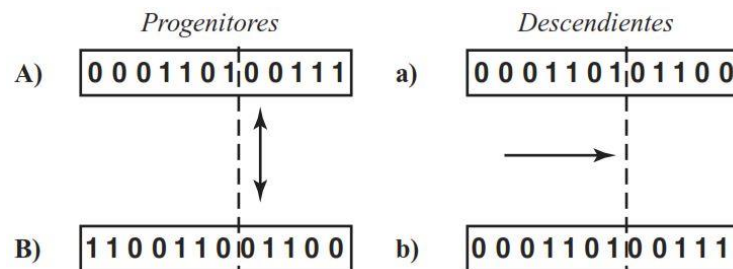


Gráfico 2.8 Cruzamiento de único-punto.
Elaborado por: Alfaomega Grupo Editor, S.A

Para otros problemas o diferentes codificaciones, otros métodos de cruzamiento pueden ser útiles o incluso necesarios. Se mencionan algunos de ellos a continuación.

- Cruzamiento de N-puntos: en lugar de un único punto, se eligen al azar N puntos de ruptura. Cada segunda sección se intercambia. Entre estas clases, la de dos-puntos es particularmente importante.
- Cruzamiento segmentado: esta técnica es muy parecida al cruzamiento de N-puntos, con la diferencia de que el número de puntos de ruptura puede variar.

- Cruzamiento uniforme: para cada posición, se decide al azar si se intercambian las posiciones.
- Cruzamiento aleatorio: primero se escoge una permutación aleatoria que se aplicará a los progenitores, después el cruzamiento de N-puntos se aplica a los progenitores aleatorios, y finalmente los descendientes aleatorios son transformados de nuevo con permutación inversa [18].

3. Mutación

El último ingrediente de los AG es la mutación, esto es la deformación aleatoria de la información genética en un individuo como las radiaciones radioactivas u otros medios de influencia. En la reproducción real, la probabilidad de que determinado gen sea mutado es casi igual para todos los genes. Así, está al alcance de la mano usar las siguientes técnicas de mutación para una determinada cadena binaria s , donde p_M es la probabilidad de que un solo gen sea modificado:

```

Algoritmo
FOR i:= 1 TO n DO
IF Random[0,1] <  $p_M$  THEN
Invert  $s[i]$ ;

```

Por supuesto que se pueden encontrar muchas alternativas y con más detalles. Algunas de estas técnicas se muestran a continuación.

- Inversión de un solo bit: la probabilidad de mutación p_M de que un bit elegido al azar sea negado.
- Inversión por fragmentos: toda la cadena es invertida bit a bit con una probabilidad de mutación p_M .
- Selección aleatoria: la probabilidad de mutación p_M de que una cadena elegida al azar sea remplazada [18].

4. Remplazo

Procedimiento para calcular (crear) una nueva generación de la anterior y sus descendientes. Se crea un espacio a la descendencia en la población eliminando de ella a los padres. Los algoritmos genéticos trabajan con un número fijo de cadenas binarias de longitud fija. Para este fin, se asumirá que las cadenas que se va a considerar son del mismo conjunto

$$S = \{0,1\}^n;$$

Por lo tanto, la generación en el tiempo t es una lista de m cadenas donde se denotará como [18]:

$$B_1 = (b_{1,t}, b_{2,t}, \dots, b_{m,t})$$

2.2.17. API

API (Application Programming Interface) es una especificación, en un lenguaje de programación, de las propiedades de un módulo de software. Los clientes del módulo solo deben depender exclusivamente de las propiedades definidas por el API de forma explícita [21].

2.2.18. Distributivo

Art. 2.- El Distributivo de Trabajo será elaborado con sujeción a las disposiciones del estatuto universitario en cada Unidad Académica (Facultad), en base a las propuestas elaboradas por las Carreras y se presentará a Consejo Académico Universitario al menos con treinta días de anticipación a la iniciación del nuevo semestre [22].

2.3. Propuesta de Solución

La realización del Módulo de Control de Distributivos y Generación de Horarios integrado al Sistema de Control de Docentes de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial permitirá el manejo de la información de los Distributivos de una manera eficaz y confiable además de la generación de Horarios de docentes; la generación de reportes de los distributivos horizontal y vertical así como también la modificación de la información para evitar datos erróneos.

CAPÍTULO III

METODOLOGÍA

3.1. Modalidad de la Investigación

En el presente proyecto se aplicó diferentes tipos de investigación, los cuales son esenciales para la mejorar los diferentes procesos además de adquirir conocimientos nuevos con la investigación.

Investigación aplicada

Se aplicará investigación aplicada porque se pondrá en práctica los conocimientos adquiridos con lo cual se llegará a resultados óptimos los cuales beneficiaran a la institución e incluso a los docentes.

Investigación de campo

Se basará en una investigación de campo porque se recompilará información necesaria y relevante para ser analizada y la fuente consultada es el administrador de Redes quien está a cargo del manejo de Sistema de Control de Docentes de la FISEI.

Investigación Bibliográfica - Documental

Se fundamentará en una investigación bibliográfica – documental porque se reunirá información similar en la cual basarse y esas fuentes serán libros, papers, revistas y tesis.

3.2. Recolección de Información

En el proyecto se utilizará la técnica de la entrevista para poder reunir información, la misma que será realizada el Departamento de Redes – FISEI al Administrador de Redes

3.2.1 Preguntas Entrevista Subdecano

Pregunta N° 1

¿En qué lineamientos se basan para elaborar los Distributivos?

Pregunta N° 2

¿Cuál es el proceso para elaborar los Distributivos?

Pregunta N°3

¿En qué se almacena la información de los Distributivos?

Pregunta N° 4

¿Los Distributivos se encuentran disponibles de forma automatizada?

3.2.2 Preguntas Entrevistas Coordinador de Carrera

Pregunta N° 1

¿Qué requerimientos son necesarios para la elaboración de horarios?

Pregunta N° 2

¿Cómo se maneja la información de la Disponibilidad de Docentes?

Pregunta N°3

¿Cómo se recepta la disponibilidad de los Docentes?

Pregunta N° 4

¿Cuál es el proceso para la elaboración de Horarios?

3.2.3 Preguntas Entrevistas Administrador de Redes

Pregunta N° 1

¿Cómo se maneja el Sistema de Control de Docentes en cuanto a Horarios?

Pregunta N° 2

¿Cómo se almacena la información de Horarios?

Pregunta N°3

¿Qué tipos de reportes genera el sistema en Cuanto a Distributivos y Horarios?

Pregunta N° 4

¿Qué problemas han existido con la forma de ingreso de Horarios?

Pregunta N° 5

¿Cree usted que sería necesario la generación de Horarios de forma automatizada?

3.3. Procesamiento y análisis de datos

La entrevista se realizó para reunir información acerca de los procesos los distributivos tanto horizontal como vertical y de la realización de horarios que se llevan a cabo en la FISEI, lo mismo que fue base fundamental para el desarrollo del Módulo de Control de Distributivos y Generación de Horarios Integrado al Sistema de Control de Docentes de la FISEI en la Universidad Técnica de Ambato.

3.3.1 Proceso Preguntas Entrevista Subdecano

Pregunta N° 1

¿En qué lineamientos se basan para elaborar los Distributivos?

Respuesta

Se basa en varios aspectos entre ellos:

El cumplimiento que para un docente a tiempo completo debe estar entre 15 a 19 horas por que tenemos asignaturas que tienen créditos impares (el del CES habla de 16 a 20 horas).

El número de horas que debe cumplir como tutor: tutorías académicas y tutorías estudiantiles.

Complementar sus horas a tiempo completo como miembro de diferentes comisiones.

El reglamento dice por cada 4 horas de clases debe tener una de preparación de clase, eso significa que si yo tengo 20 horas clases en un caso extremo constaría 5 de preparación de clases, 2 de tutorías académicas, 3 para comisión estudiantil y si es parte de un proyecto de investigación son 8 horas por ley para investigación y las dos que faltan se podría poner por ejemplo 2 de tutorías de tesis. Es importante saber que todos estos lineamientos vienen por parte de la DEAC y emanadas desde vicerrectorado académico.

Pregunta N° 2

¿Cuál es el proceso para elaborar los Distributivos?

Respuesta

En nuestro caso se les encarga a los coordinadores de carrera de acuerdo a las necesidades de estas se realiza una distribución tanto en horas como en asignaturas de acuerdo a las especialidad de cada docente buscando que estos no vayan a dar más allá de 4 módulos de asignaturas. Se acentúa un distributivo tanto horizontal y vertical para cotejar las asignaturas entregadas a cada Docente así como sus respectivas comisiones.

Pregunta N°3

¿En qué se almacena la información de los Distributivos?

Respuesta

La información se encuentra almacenada en medios magnéticos y reposan en cada una de las carreras de manera individualizada y el compilado de la facultad reposa en secretaria del subdecanato.

Pregunta N° 4

¿Los Distributivos se encuentran disponibles de forma automatizada?

Respuesta

No se dispone de forma automatizada para la elaboración de distributivos.

Análisis

El problema se encuentra en que la información de Distributivos esta almacenada de forma individualizada y no está integrada en el Sistema de Control de Docentes lo cual no permite generar un reporte de manera rápida.

3.3.2 Proceso Preguntas Entrevistas Coordinador de Carrera

Pregunta N° 1

¿Qué requerimientos son necesarios para la elaboración de horarios?

Respuesta

Primero contar con el distributivo de los Docentes, en la distribución debe de estar los módulos refinados, las HAC que son las Áreas académico si tiene dirección de tesis o si está en una de las 12 comisiones que manejan. Una vez establecido eso el segundo parámetro seria el horario de comisiones sensibles (Comisiones en donde todos los integrantes tienen que coincidir el horario) las comisiones que tiene que coincidir el

horario de ley para trabajar en conjunto son Rediseño Curricular y la de la Unidad de Planificación y Evaluación (UPE) esas dos comisiones deben de estar siempre en la misma carga horaria.

Saber las reuniones de consejo académico, consejo directivo, consejo académico universitario y consejo universitario de tal manera que las personas que participan ahí ya tiene un horario preestablecido por ejemplo aquí también tendría que determinar qué días se van a reunir para que ahí tengan en cuenta esa carga horaria y no se le cruce por ejemplo un profesor que está dando clase siempre va estar faltando a clases otra situación dentro de eso la disponibilidad seria lo último, la disponibilidad de la carga horaria que tiene el profesor eso tomando en cuenta que hay una disposición que debe ser 5/3 o 4/4 ningún profesor puede trabajar más de 5 horas en 1 banda horaria de la mañana a la tarde está dentro del lineamiento, todos los lineamientos para el distributivos salen cada semestre ya que se elaboran los distributivos, en base a eso y que hay materias que son muy importantes sería en lo posible que nunca se pongan los días viernes porque generalmente los viernes se pierden y perder un viernes es muy importante y repercute en que los alumnos podrían perder clases entonces hay ciertas materias que no deberían ponerse los viernes a parte habrían que ver si esas materias son de laboratorio.

Otro análisis seria que materias es importante que reciban en la mañana y que materias en la tarde también podría ser una ayuda al estudiante porque hay materias muy complejas, capaz que se le ponen en la última carga horaria y no puede recibir adecuadamente y obviamente el otro requisito es la disponibilidad de laboratorios porque lo anteriormente dicho seria secundario si es que realmente hay un choque ya no se le podría poner pero también podría ponerse en consideración en caso de que exista laboratorios disponibles se podría tomar en cuenta ese tipo de materias que si es compleja, es practica o netamente teórica.

Después de todos esos requisitos mencionados anteriormente ahí vendría a ser la realización del horario.

Pregunta N° 2

¿Cómo se maneja la información de la Disponibilidad de Docentes?

Respuesta

Los docentes entregan un horario de su disponibilidad. Y la finalidad con eso es ayudarles de alguna forma a los docentes pero se considera siempre y cuando haya disponibilidad en el horario ósea que se pueda acomodar en el horario de clases pero no es un documento oficial es más por ayuda.

Pregunta N°3

¿Cómo se recepta la disponibilidad de los Docentes?

Respuesta

Se imprime unas hojas con formatos de horarios pequeños además se les envía un correo electrónico en el que se les comunica que venga a dejar su disponibilidad y los docentes vienen a entregar en secretaria de subdecanato.

Pregunta N° 4

¿Cuál es el proceso para la elaboración de Horarios?

Respuesta

La parte principal son dar pesos a los laboratorios como prioridad de uso a que me refiero con eso que primero se asigna las horas que necesitan ocupar laboratorios y luego vamos asignando las horas que serían de relleno de cada horario se arma el horario del curso y se va armando a la par el horario de los docentes que se involucran con ese curso por nivel como repito la parte fundamental es los laboratorios porque si yo cuadro los horarios sin pensar en laboratorios va a ver muchos cruces tomando en cuenta que tenemos 2 laboratorios para cubrir 3 carreras en la parte básica que es el laboratorio 1y 2 para TICS y programación y dentro de eso le involucramos alrededor

de 4 o 5 docentes para cubrir esas materias eso se tiene que tomar en cuenta siempre como primer paso.

Segundo también priorizamos los requerimientos de la universidad que son no contemplar más de 5 horas clase y además procuramos se impartan materias de mucho peso seguidas ejemplo procuramos no poner Geometría, Álgebra, Física, Cálculo en el mismo día tratamos de romper esos ciclos, cuando no se puede se pondrá una materia que sea menos pesada que no involucre números como son Técnicas de Estudio, Lenguaje y Comunicación o TICS en medio de 4 horas de materias que son números por ejemplo 2 de Física, 1 de TICS y terminamos con 2 de Geometría.

Y de igual forma tratamos en lo posible de los requerimientos que los docentes nos dan como digamos horas que ellos no pueden venir a dar clase o no pueden cumplir con la facultad porque tienen otras prioridades anterior a eso, cuáles son esas prioridades talvez tener inglés o talvez no quieren tener clases desde las 7 de la mañana porque es más cómodo para ellos por cuestiones de esas se trata también de cubrir las necesidades sin ser eso una camisa de fuerza. Se trata de cumplirle al docente pero también hay que recordar que dentro de la universidad cuando uno firma un contrato para ellos dice que está inmerso a cumplir el horario que la universidad y la facultad lo requiera al docente no que el docente le ponga horario a la facultad creo que ese sería el proceso fundamental. Dentro de ese proceso se lo realiza de forma manual y luego se sube al sistema por eso es que necesitamos la automatización de forma que en el momento de subir al sistema se va viendo a veces los cruces que si se dan porque es un margen de error humano que siempre van a tener las personas de administración de redes que somos 8 personas.

Análisis:

La información necesaria para la elaboración de horarios como los distributivos y estos no se encuentra almacenada en una base de datos, y es necesario recalcar que los distributivos son esenciales porque en ellos se basa la mayoría de información requerida para los horarios; además la disponibilidad de los docentes se maneja de forma separada no permitiendo que el Sistema pueda manejar esa información que es de vital

importancia para los horarios y no permitiendo por estas razones un manejo más óptimo de la creación de horarios retrasando su proceso.

3.3.3 Proceso Preguntas Entrevistas Administrador de Redes

Pregunta N° 1

¿Cómo se maneja el Sistema de Control de Docentes en cuanto a Horarios?

Respuesta

El Sistema control docente requiere que sea alimentado con datos de horarios ya preestablecidos de forma manual, el sistema trabaja con una tabla de horarios actual se llena mediante las tablas de docente, niveles, catálogos que es la que maneja un distributivo de forma se podría decir rustica porque se ingresa el docente, la materia que da y el número de horas que se asigna a cada materia o a cada hora complementaria.

Pregunta N° 2

¿Cómo se almacena la información de Horarios?

Respuesta

Dentro de la base de datos en la tabla horarios se hace realmente un insert directo y se almacenan los horarios ahí.

Pregunta N°3

¿Qué tipos de reportes genera el sistema en Cuanto a Distributivos y Horarios?

Respuesta

No tenemos reportería en cuanto a distributivos porque el sistema no tiene un módulo de Distributivos; en cuanto horarios se generan digamos que más como reportes es una impresión del horario por docente y el horario por semestre aunque ya esa entidad de

semestre no es trabajable dentro de la Universidad porque se está usando créditos pero todavía se maneja el esquema del primer semestre paralelo.

Pregunta N° 4

¿Qué problemas han existido con la forma de ingreso de Horarios?

Respuesta

El problema creo que es un poco más que del sistema es de tipeo o de sintaxis que uno vaya llevando en la parte de carga de horarios, nos equivocamos al subir los horarios en horas por ejemplo si tenía clases de matemáticas de 7 a 8 y nos equivocamos y lo pusimos de 8 a 9 o no estaba asignado a primero y nosotros le asignamos a primero. Son errores que tiene son más de usuario que del sistema.

Pregunta N° 5

¿Cree usted que sería necesario la generación de Horarios de forma automatizada?

Respuesta

Creo que sí, la utilidad de tener horarios de una forma automática va a viabilizar y facilitar el trabajo. Nos va a dar un borrador de los horarios de los cuales tendríamos que irlos manipulando para hacer cambios mas no generara un horario desde cero.

Análisis:

El Sistema de Control Docente no genera un reporte de Distributivos porque el modulo no está integrado en el sistema ni la información se encuentra en la base de datos. Por ello es necesario la elaboración del módulo así como la integración de la información de disponibilidad permitiendo que de esta manera se pueda realizar de forma automatizada la generación de horarios siendo un gran aporte y ayuda para los involucrados en estos procesos.

3.4. Desarrollo del proyecto

1. Analizar el funcionamiento de la base de datos del Sistema de Control de Docentes.

- Análisis de las relaciones existentes en la base de datos del Sistema de Control de Docentes de la FISEI.
- Integración de las relaciones necesarias en la base de datos del Sistema de Control de Docentes de la FISEI.
- Reestructuración de la base de datos del Sistema de Control de Docentes de la FISEI.
- Implementación de los cambios necesarios en la base de datos existe del Sistema de Control de Docentes para acoplarse a la modalidad de créditos de la UTA.

2. Diseñar el Módulo de Control de Distributivos y Generación de Horarios en base al levantamiento de requerimientos.

- Análisis del front-end del Sistema de control de Docentes de la FISEI.
- Definición de requerimientos indispensables para el funcionamiento del Módulo conjuntamente con el Sistema.

3. Desarrollar el Módulo de Control de Distributivos y Generación de Horarios.

- Implementación de los controles y Seguridades para el Sistema de Control de Docentes conjuntamente con el Módulo de Control de Distributivos y Generación de Horarios acoplado.

4. Pruebas del Módulo de Control de Distributivos y Generación de Horarios.

- Realización de Pruebas de Funcionamiento.
- Integración del Módulo de Control de Distributivos y Generación de Horarios, en sí, el sistema completo.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1. Tema

Módulo de Control de Distributivos y Generación de Horarios Integrado al Sistema de Control de Docentes de la FISEI en la Universidad Técnica de Ambato.

4.2. Datos Informativos

Institución ejecutora: Departamento de Redes en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

Dirección: Av. Los Chasquis entre Rio Payamino y Rio Guayllabamba (Campus Huachi Chico, Segundo piso). Teléfono: 032851894 Ext. 118.

Beneficiarios: Facultades de la Universidad Técnica de Ambato.

Tiempo: El presente proyecto se lo realizará entre el 12 de Febrero de 2015 y el 30 de marzo de 2016.

Costo: El costo estimado para la realización de este proyecto es de \$ 7048,97.

Tutor: Ing. Franklin Mayorga, Mg.

4.3. Antecedentes de la propuesta

El Departamento de Redes de la FISEI es el encargado del Sistema de Control Docente de la Facultad, de la parte de Redes de la facultad además de los laboratorios que en ella existe y de los equipos que los estudiantes utilizan para hacer las prácticas en las diferentes asignaturas. El Departamento también se encargan de la elaboración de horarios antes de empezar el siguiente periodo académico aunque este proceso ha causado algunos inconvenientes al momento de subir los horarios elaborados a mano con algunos docentes. Y con la cantidad de materias y laboratorios que se maneja se genera pérdida de tiempo e inconsistencia en algunos horarios.

Por los problemas anteriormente identificados surge la necesidad de un Módulo de Control de Distributivos y Generación de Horarios Integrado al Sistema de Control de Docentes de la FISEI en la Universidad Técnica de Ambato, el mismo que estará basado en los requerimientos del Departamento de redes, lo cual hará confiable los datos obtenidos en los Horarios.

La Universidad Técnica De Ambato cuenta con licencias las cuales será tomada en cuenta para elegir las herramientas más aptas para el desarrollo de dicho módulo.

4.4. Justificación

Mediante el desarrollo del presente módulo se podrá satisfacer las necesidades previamente establecidas en cuanto a la automatización de los procesos de control de distributivos y generación de horarios, evitando inconvenientes con la información que maneja el sistema e incluso la optimización del tiempo en cuanto a la generación de la información de los horarios de los docentes Anexo 4.

El Módulo Control de Distributivos y Generación de Horarios permitirá que en cada periodo académico se den mejoras en la generación de horarios buscando cada vez más su optimización. Obteniendo así horarios confiables que cumplan con las normas necesarias.

4.5. Análisis de Factibilidad

4.5.1 Factibilidad Operativa

El Módulo se realizará median los requerimientos establecidos en la información que se obtuvo de las encuestas aplicadas a el Administrador de Redes, Subdecano y Coordinador de Carrera; de modo que el módulo será amigable y de fácil comprensión para los usuarios involucrados en el procesos de Distributivos y Horarios además que el modulo se puede adaptar para que funcione en diferentes facultades todo esto ser posible mediante roles previamente determinados. De modo que los usuarios que lo utilicen puedan adaptarse con tranquilidad siendo de gran ayuda para mantener disponible y segura la información de los procesos expuestos. Después de haber expuesto todo lo anteriormente dicho se ha llegado a la conclusión de que el módulo es factible desde la perspectiva operativa.

4.5.2 Factibilidad Económica

Mediante el respaldo obtenido por la Administración de Redes y la FISEI, se puede emplear las licencias necesarias en cuanto a software, además de servidores, acceso a internet e incluso a la información requerida para la realización de este módulo.

Dentro de las herramientas de desarrollo tenemos SQL Server 2008 R2, Visual Studio 2012, las mismas que fueron proporcionadas por la Administración de Redes por su facilidad de manejo, por el soporte de varios lenguajes, por sus múltiples beneficios, por su rendimiento y la calidad que ofrecen además de las licencias que previamente han sido adquiridas.

A continuación se describe los costos de los recursos necesarios para el desarrollo del Módulo:

RECURSOS HUMANOS			
Nº	CARGO	VALOR UNITARIO	VALOR TOTAL
1	Analista	900	900
1	Programador	800	800
1	Investigador	1000	1000
TOTAL			2700

Tabla 3.1 Recursos Humanos
Elaborado por: Andrea Sotelo

RECURSOS TECNOLÓGICOS				
HARDWARE				
DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Computadora	c/u	1	900	900
Servidor	c/u	1	460	460
SOFTWARE				
DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Licencia SQL Server 2008	c/u	1	500	500
Licencia Visual Studio 2013	c/u	1	500	500
Licencia Microsoft Office	c/u	1	300	300
			TOTAL	2660

Tabla 3.2 Recursos Tecnológicos
Elaborado por: Andrea Sotelo

RECURSOS MATERIALES				
DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Copias	c/u	80	0,02	1,60
Impresiones b/n	c/u	300	0,10	30,00
Impresiones color	c/u	100	0,25	25,00
Papel bond A4	c/u	500	0,05	25,00
Esferos	c/u	3	0,35	1,05
Lápiz	c/u	2	0,25	0,50
Transporte	veces	500	0,30	150,00
Internet	horas	1000	0,80	800
Memoria USB Kingston 16GB	c/u	1	15,00	15,00
			Total	1048,15

Tabla 3.3 Recursos Materiales
Elaborado por: Andrea Sotelo

FLUJO DE PAGOS	
RECURSOS	VALOR
Recursos Humanos	2700
Recursos Tecnológicos	2660
Recursos Materiales	1048,15
Imprevistos (10%)	640,82
TOTAL	7048,97

Tabla 3.4 Flujo de Pagos
Elaborado por: Andrea Sotelo

De los recursos expuestos son varios los que nos proporciona la Administración de Redes y la FISEI entre ellos tenemos:

RECURSOS TECNOLÓGICOS				
HARDWARE				
DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Computadora	c/u	1	900	900
Servidor	c/u	1	460	460
SOFTWARE				
DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Licencia SQL Server 2008	c/u	1	500	500
Licencia Visual Studio 2013	c/u	1	500	500
Licencia Microsoft Office	c/u	1	300	300
			TOTAL	2660

RECURSOS MATERIALES				
DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Internet	horas	1000	0,80	800
			Total	800

Cubriendo un total de

FLUJO DE PAGOS	
RECURSOS	VALOR
Recursos Tecnológicos	2660
Recursos Materiales	800
TOTAL	3460

Y los recursos restantes son proporcionados por el investigador entre ellos tenemos:

RECURSOS HUMANOS			
Nº	CARGO	VALOR UNITARIO	VALOR TOTAL
1	Analista	900	900
1	Programador	800	800
1	Investigador	1000	1000
		TOTAL	2700

RECURSOS MATERIALES				
----------------------------	--	--	--	--

DETALLE	UNIDAD	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Copias	c/u	80	0,02	1,60
Impresiones b/n	c/u	300	0,10	30,00
Impresiones color	c/u	100	0,25	25,00
Papel bond A4	c/u	500	0,05	25,00
Esferos	c/u	3	0,35	1,05
Lápiz	c/u	2	0,25	0,50
Transporte	veces	500	0,30	150,00
Memoria USB Kingston 16GB	c/u	1	15,00	15,00
			Total	248,15

Cubriendo un total de

FLUJO DE PAGOS	
RECURSOS	VALOR
Recursos Humanos	2700
Recursos Materiales	248,15
Imprevistos (10%)	640,82
TOTAL	3588,97

Conclusión:

Se ha demostrado que es factible aplicar el presente proyecto porque la Administración de Redes y la FISEI además del investigador proporcionan lo necesario para el desarrollo del mismo.

4.5.3 Factibilidad Técnica

Con la Implementación del Módulo de Control de Distributivos y Generación de Horarios Integrado al Sistema de Control de Docentes de la FISEI en la Universidad Técnica de Ambato, se quiere mejorar el manejo de la información además del ahorro de tiempo en cuanto a los procesos respectivos para Distributivos y Horarios lo que beneficia a la Facultad.

El Modulo se alojara en el servidor en el que se encuentra el Sistema de Control de Docentes de la Facultad de Ingeniería en Sistemas, electrónica e Industrial.

Hardware

Disco: 100 GB

RAM: 2GB

Sistema Operativo: Windows Server 2008

Service Pack: 1

De los cuales el Modulo consumirá un espacio no máximo a

Disco: 1GB

RAM: 2GB

Software

Lenguaje de Programación: ASP. NET

IDE de Desarrollo: MS Visual Studio

Motor de base de datos: SQL Server 2008 R2

Complementos: JQuery, Ajax, JavScript, ItextSharp

Conclusión:

Se ha demostrado que es factible aplicar la implementación del Módulo porque el Hardware y Software que la FISEI proporciona, cumple con lo requerido para que se aproveche de los beneficios del presente proyecto.

4.6. Fundamentación

ASP.NET

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc.

ASP.NET incluye:

Marco de trabajo de página y controles, Compilador de ASP.NET, Infraestructura de seguridad, Funciones de administración de estado, Configuración de la aplicación, Supervisión de estado y características de rendimiento, Capacidad de depuración, Marco de trabajo de servicios Web XML, Entorno de host extensible y administración del ciclo de vida de las aplicaciones, Entorno de diseñador extensible [23].

Visual Studio

Visual Studio 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 hace más sencilla 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 [24].

SQL Server 2008 R2

Microsoft® SQL Server™ es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos. En esta sección, encontrará información sobre varias versiones de SQL Server. También encontrará artículos sobre bases de datos y aplicaciones de diseño de bases de datos así como ejemplos de los usos de SQL Server [25].

JQuery

jQuery es considerado un Framework de Javascript, o ambiente de desarrollo. Lo que no es más que un conjunto de utilidades las cuales no necesitan ser

programadas, de hecho ya fueron programadas, probadas y podemos utilizarlas de una manera muy simplificada.

En otras palabras, podremos lograr los mismos resultados, en menos tiempo sin necesidad de programar una funcionalidad completamente [26].

JavaScript

JavaScript es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web [27].

iTextSharp

iTextSharp es una librería, de código abierto (open source) y específicamente para .Net, que nos permite crear y modificar documentos PDF. Si usted prefiere programar en Java, también se puede beneficiar de iText para Java y Android. E [28].

4.7. Metodología

Para la elaboración del Módulo se utilizó la metodología XP, XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Proceso XP

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto [29].

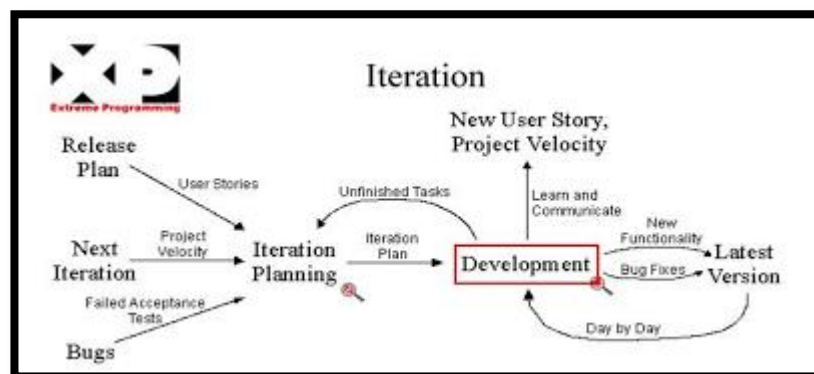


Gráfico 4.0 Ciclo de vida XP
Elaborado por: Andrea Sotelo

Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para

que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas.

- **El juego de la planificación:** Hay una comunicación frecuente el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- **Entregas pequeñas:** Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.
- **Metáfora:** El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).
- **Diseño simple:** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- **Pruebas:** La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.
- **Refactorización (Refactoring):** Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- **Programación en parejas:** Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores,...).
- **Propiedad colectiva del código:** Cualquier programador puede cambiar cualquier parte del código en cualquier momento.

- **Integración continúa:** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- **40 horas por semana:** Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.
- **Cliente in-situ:** El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- **Estándares de programación:** XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible [29].

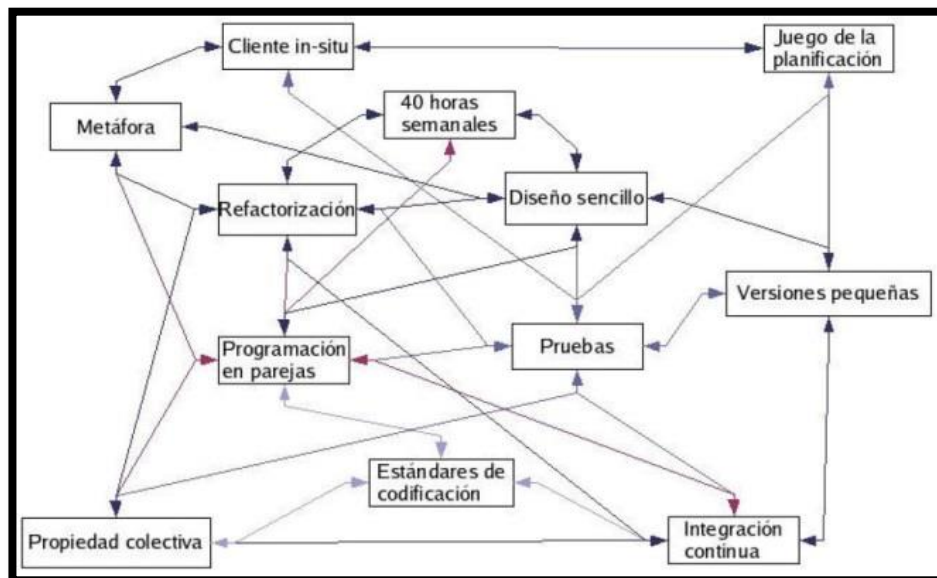


Gráfico 4.1 Prácticas XP
Elaborado por: Andrea Sotelo

4.8. Modelo Operativo

4.8.1 Requerimientos del Sistema

Para poder determinar los requerimientos necesarios se aplicó una encuesta por la cual se llegó a la conclusión que la Facultad de Ingeniería en Sistemas,

Electrónica e Industrial, no cuenta con un Módulo de Control de Distributivos y generación de Horarios, por lo que se emplea más tiempo al realizar horarios y la información de los distributivos no se encuentra segura.

Para el desarrollo del presente Modulo, se identificaron los requerimientos necesarios con el Administrador de Redes para automatizar los procesos tanto de Horarios como de Distributivos, y se determinó que en el sistema deberá tener:

- Administrar de Usuarios
- Administrar de Aulas
- Administrar de Malla
- Disponibilidad Docente
- Distributivos
- Generación de Horarios
- Generar Reportes

4.8.2 Diagramas UML

Es un lenguaje para modelar sistemas de software el cual se basa en el análisis y diseño orientados a objetos. Este es lenguaje grafico para construir, documentar, visualizar y especificar un sistema.

UML consta de varios beneficios entre ellos tenemos:

- Mejores tiempos totales de desarrollo.
- Modelar sistemas utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

4.8.3 Diagramas de casos de usos

El siguiente diagrama se elaboró con el equipo de desarrollo y el administrador de Redes de la FISEI además de basarse en los requerimientos establecidos.

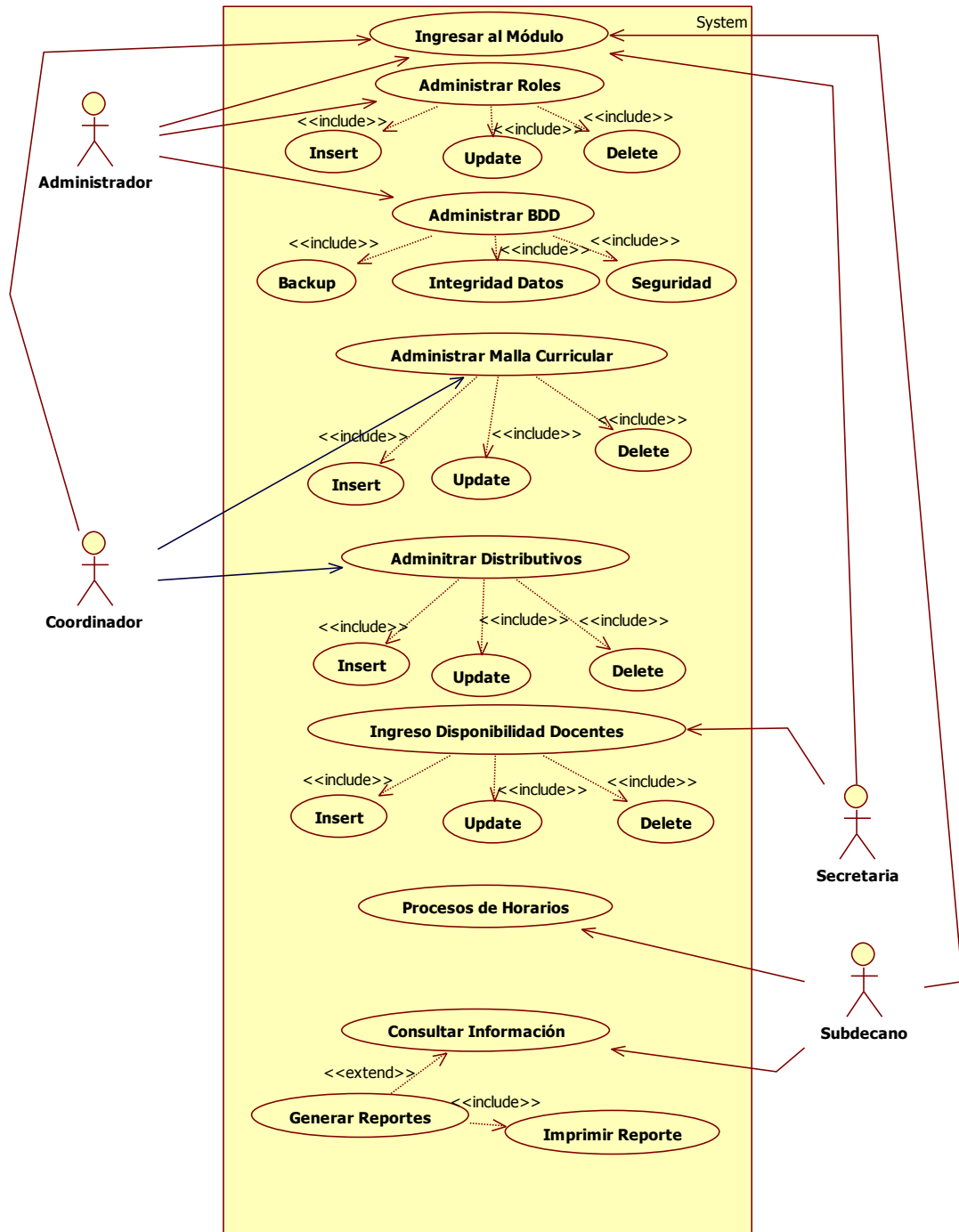


Gráfico 4.2 Diagrama Caso de Uso de los Procesos del Módulo
Elaborado por: Andrea Sotelo

4.8.4 Especificaciones de casos de usos

Caso de Uso: Ingreso al Módulo	
Código: AR-0001	
Actores:	<ul style="list-style-type: none">➤ Administrador➤ Coordinador➤ Subdecano➤ Secretaria
Descripción:	<ul style="list-style-type: none">➤ Faculta el acceso al sistema para realizar actividades que sean autorizadas según su rol.
Precondiciones:	<ul style="list-style-type: none">➤ Usuario registrado con su rol asignado.
Post condición:	<ul style="list-style-type: none">➤ Ninguna
Flujo Principal:	<ol style="list-style-type: none">1. Ejecuta al Módulo2. Ingresa el usuario y la contraseña3. Hace clic en el botón Ingresar
Flujo Alternativo:	2.1 Si la información está mal ingresada aparecerá un mensaje de error: Error! Su contraseña no es correcta o no es de Administrador

Tabla 4.0 Caso de Uso: Ingreso al Módulo
Realizado por: Andrea Sotelo

Caso de Uso: Ingreso de Información de Malla Curricular
Código: AR-0008
Actores: <ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador
Descripción: <ul style="list-style-type: none"> ➤ Ingreso de la información necesaria para la malla de cada carrera
Precondiciones: <ul style="list-style-type: none"> ➤ Materia Registrada ➤ Nivel Registrado
Post condición: <ul style="list-style-type: none"> ➤ Elaboración de la malla de cada carrera con la información necesaria
Flujo Principal: <ol style="list-style-type: none"> 1. Seleccionar en el Menú Malla 2. Ingresar tipo de malla 3. Seleccionar la Materia 4. Seleccionar la Carrera 5. Seleccionar el nivel 6. Seleccionar el periodo 7. Hace clic en el botón Guardar
Flujo Alternativo: 2.1 Si la información está mal ingresada aparecerá un mensaje de error: Debe Ingresar Tipo Debe ingresar Horas

Tabla 4.1 Caso de Uso: Ingreso de Información de Malla Curricular
Realizado por: Andrea Sotelo

Caso de Uso: Modificación de Información de Malla Curricular
Código: AR-0009
Actores: <ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador
Descripción: <ul style="list-style-type: none"> ➤ Modificar la información de la malla que fue ingresada con algún error.
Precondiciones: <ul style="list-style-type: none"> ➤ Información de Malla Registrada
Post condición: <ul style="list-style-type: none"> ➤ Información de malla actualizada de manera correcta.
Flujo Principal: <ol style="list-style-type: none"> 1. Seleccionar en el Menú Malla 2. Seleccionar la información de malla que se va a modificar 3. Actualizar el la información de la Malla 4. Guardar los cambios realizados.

Tabla 4.2 Caso de Uso: Modificación de Información de Malla Curricular
Realizado por: Andrea Sotelo

Caso de Uso: Eliminación de Información de Malla Curricular	
Código: AR-0010	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador
Descripción:	<ul style="list-style-type: none"> ➤ Eliminar el la información del registro de la malla.
Precondiciones:	<ul style="list-style-type: none"> ➤ Información de malla registrada que no esté siendo ocupada en los distributivos.
Post condición:	<ul style="list-style-type: none"> ➤ Información de la malla Eliminada
Flujo Principal:	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Malla 2. Seleccionar la información de la Malla que se va a eliminar 3. Hacer Clic en el botón eliminar

Tabla 4.3 Caso de Uso: Eliminación de Información de Malla Curricular
Realizado por: Andrea Sotelo

Caso de Uso: Ingreso de la Información de Distributivos	
Código: AR-0011	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador
Descripción:	<ul style="list-style-type: none"> ➤ Ingreso de la información necesaria para los distributivos de cada carrera
Precondiciones:	<ul style="list-style-type: none"> ➤ Usuario Registrada ➤ Periodo Registrado ➤ Información de la Malla Registrada
Post condición:	<ul style="list-style-type: none"> ➤ Elaboración del Distributivo de cada carrera con la información necesaria
Flujo Principal:	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Distributivo 2. Seleccionar Docente/Usuario 3. Seleccionar Periodo 4. Seleccionar la Información de la Malla 5. Ingresar paralelo 6. Ingresar horas 7. Ingresar cupo 8. Seleccionar Estado 9. Hace clic en el botón Guardar
Flujo Alternativo:	<p>2.1 Si la información está mal ingresada aparecerá un mensaje de error: Debe ingresar Paralelo Debe ingresar Cupo</p>

Tabla 4.4 Caso de Uso: Ingreso de la Información de Distributivos
Realizado por: Andrea Sotelo

Caso de Uso: Modificación de la Información de Distributivos	
Código: AR-0012	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador
Descripción:	<ul style="list-style-type: none"> ➤ Modificar la información del Registro del Distributivo que fue ingresada con algún error.
Precondiciones:	<ul style="list-style-type: none"> ➤ Información del Distributivo Registrado
Post condición:	<ul style="list-style-type: none"> ➤ Información del Distributivo actualizado de manera correcta.
Flujo Principal:	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Distributivo 2. Seleccionar la información del Distributivo que se va a modificar. 3. Ingresar Observación 4. Ingresar Detalle de la Reforma 5. Actualizar la información del Distributivo 6. Guardar los cambios realizados.

Tabla 4.5 Caso de Uso: Modificación de la Información de Distributivos
Realizado por: Andrea Sotelo

Caso de Uso: Eliminación de Información de Distributivo	
Código: AR-0013	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador
Descripción:	<ul style="list-style-type: none"> ➤ Eliminar el la información del registro del Distributivo.
Precondiciones:	<ul style="list-style-type: none"> ➤ Información del Distributivo registrado que no esté siendo ocupada en los horarios.
Post condición:	<ul style="list-style-type: none"> ➤ Información del Distributivo Eliminado.
Flujo Principal:	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Distributivo. 2. Seleccionar la información del Distributivo que se va a eliminar 3. Hacer Clic en el botón eliminar

Tabla 4.6 Caso de Uso: Eliminación de Información de Distributivo
Realizado por: Andrea Sotelo

Caso de Uso: Ingreso de Disponibilidad de Docentes	
Código: AR-0014	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Secretaria
Descripción:	<ul style="list-style-type: none"> ➤ Ingresa las horas de cada día de la semana disponible por los docentes.
Precondiciones:	<ul style="list-style-type: none"> ➤ Usuario
Post condición:	<ul style="list-style-type: none"> ➤ Información necesaria para la generación de los horarios de cada carrera
Flujo Principal:	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Disponibilidad 2. Seleccionar la hora 3. Seleccionar el día 4. Seleccionar la Jornada 5. Hace clic en el botón Guardar
Flujo Alternativo:	<p>2.1 Si la información está mal ingresada aparecerá un mensaje de error: No se pudo guardar los datos en el Servidor</p>

Tabla 4.7 Caso de Uso: Ingreso de Disponibilidad de Docentes
Realizado por: Andrea Sotelo

Caso de Uso: Modificación de Disponibilidad de Docentes	
Código: AR-0015	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Secretaria
Descripción:	<ul style="list-style-type: none"> ➤ Modificar la información de la Disponibilidad que fueron ingresadas con algún error de tipeo.
Precondiciones:	<ul style="list-style-type: none"> ➤ Información de Disponibilidad Registrada
Post condición:	<ul style="list-style-type: none"> ➤ Información de Disponibilidad actualizada de manera correcta.
Flujo Principal:	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Disponibilidad. 2. Seleccionar la información de Disponibilidad que se va a modificar. 3. Actualizar la Información de Disponibilidad. 4. Guardar los cambios realizados.

Tabla 4.8 Caso de Uso: Modificación de Disponibilidad de Docentes
Realizado por: Andrea Sotelo

Caso de Uso: Eliminación de Disponibilidad de Docentes	
Código: AR-0016	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Secretaria
Descripción:	<ul style="list-style-type: none"> ➤ Eliminar la información de la Disponibilidad seleccionado.
Precondiciones:	<ul style="list-style-type: none"> ➤ Información de Disponibilidad registrada que no esté siendo ocupada para la generación de horarios.
Post condición:	<ul style="list-style-type: none"> ➤ Disponibilidad Eliminada.
Flujo Principal:	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Disponibilidad. 2. Seleccionar la información de la Disponibilidad que se va a eliminar 3. Hacer Clic en el botón eliminar

Tabla 4.9 Caso de Uso: Eliminación de Disponibilidad de Docentes
Realizado por: Andrea Sotelo

Caso de Uso: Proceso de Horarios	
Código: AR-0014	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Secretaria
Descripción:	<ul style="list-style-type: none"> ➤ Ingresas las horas de cada día de la semana disponible por los docentes.
Precondiciones:	<ul style="list-style-type: none"> ➤ Usuario
Post condición:	<ul style="list-style-type: none"> ➤ Información necesaria para la generación de los horarios de cada carrera
Flujo Principal:	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Generar horarios 2. Hace clic en el botón Generar Horarios 3. Seleccionar los Horarios 4. Guardar Horarios

Tabla 4.10 Caso de Uso: Proceso de Horarios
Realizado por: Andrea Sotelo

Caso de Uso: Consulta de Información
Código: AR-0015
Actores: <ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador ➤ Secretaria ➤ Subdecano
Descripción: <ul style="list-style-type: none"> ➤ Se podrá consultar la información según la asignación de roles.
Precondiciones: <ul style="list-style-type: none"> ➤ Información Registrada
Post condición: <ul style="list-style-type: none"> ➤ Revisar la información respectiva.
Flujo Principal: <ol style="list-style-type: none"> 1. Seleccionar en el Menú el ítem necesario. 2. Ejecuta la Consulta.

Tabla 4.11 Caso de Uso: Consulta de Información
Realizado por: Andrea Sotelo

Caso de Uso: Generar Reporte
Código: AR-0016
Actores: <ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador ➤ Secretaria ➤ Subdecano
Descripción: <ul style="list-style-type: none"> ➤ Se crea reporte de la información necesaria.
Precondiciones: <ul style="list-style-type: none"> ➤
Post condición: <ul style="list-style-type: none"> ➤ Reporte de la Información
Flujo Principal: <ol style="list-style-type: none"> 1. Seleccionar en el Menú Reportes. 2. Seleccionar tipo de Reporte 3. Hacer Clic en el botón Generar

Tabla 4.12 Caso de Uso: Generar Reporte
Realizado por: Andrea Sotelo

Caso de Uso: Imprimir Reporte	
Código: AR-0016	
Actores:	<ul style="list-style-type: none"> ➤ Administrador ➤ Coordinador ➤ Secretaria ➤ Subdecano
Descripción:	<ul style="list-style-type: none"> ➤ Se Imprimirá la información obtenida en el reporte
Precondiciones:	<ul style="list-style-type: none"> ➤ Reporte Creado
Post condición:	<ul style="list-style-type: none"> ➤ Reporte Impreso con especificaciones
Flujo Principal:	<ol style="list-style-type: none"> 4. Seleccionar en el Menú Reportes. 5. Seleccionar tipo de Reporte 6. Hacer Clic en el botón Generar 7. Hacer Clic en el botón Imprimir

Tabla 4.13 Caso de Uso: Imprimir Reporte
Realizado por: Andrea Sotelo

4.8.5 Diagrama de Secuencias

El Diagrama de secuencias muestra la forma de comunicación de los objetos que participan en la interacción así como la secuencia de mensajes intercambiando en forma ordenada.

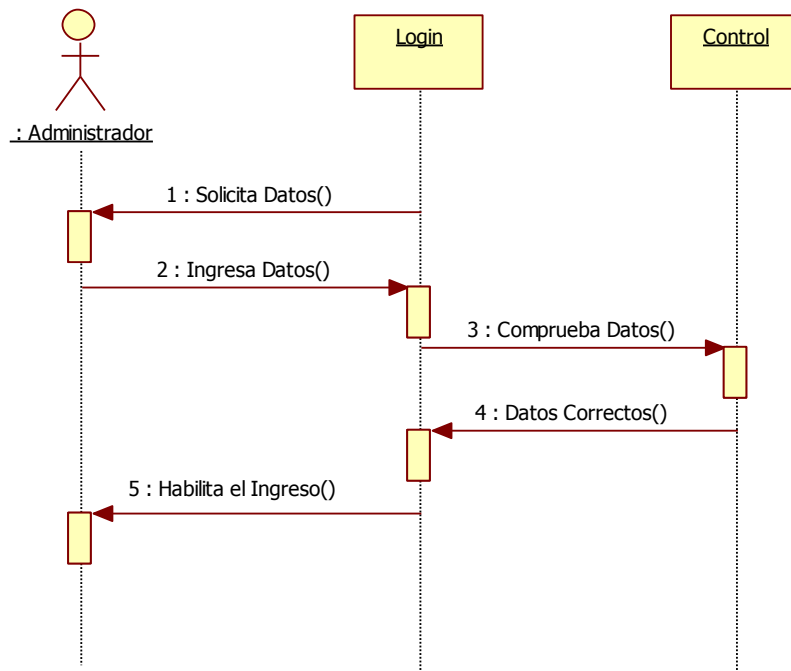


Gráfico 4.3 Diagrama de Secuencia Ingreso al Módulo
Fuente: StartUML

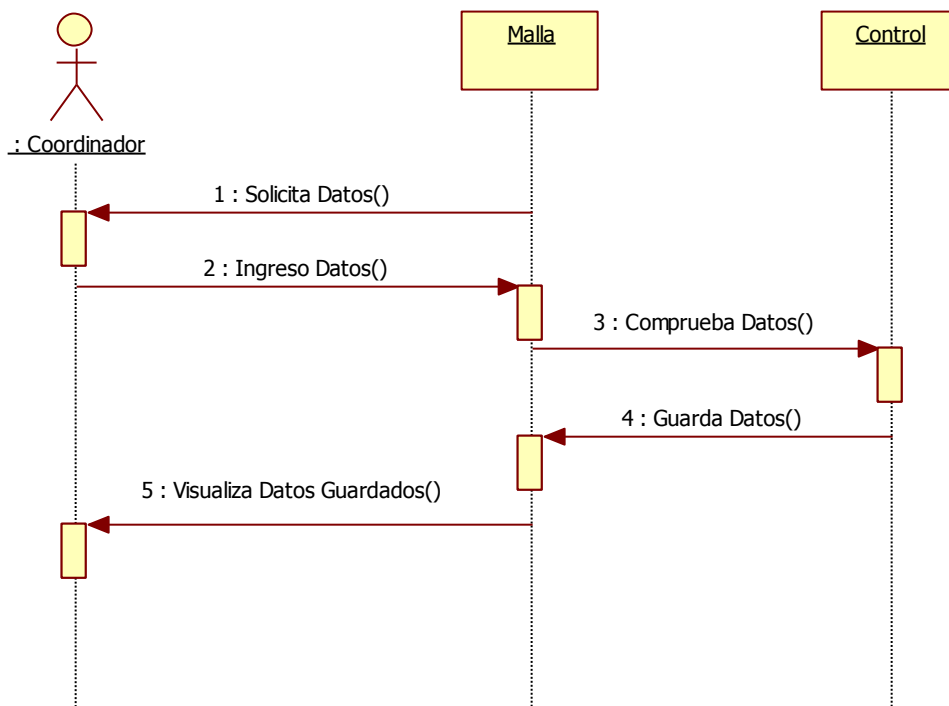


Gráfico 4.4 Diagrama de Secuencia Ingreso de Malla
Fuente: StartUML

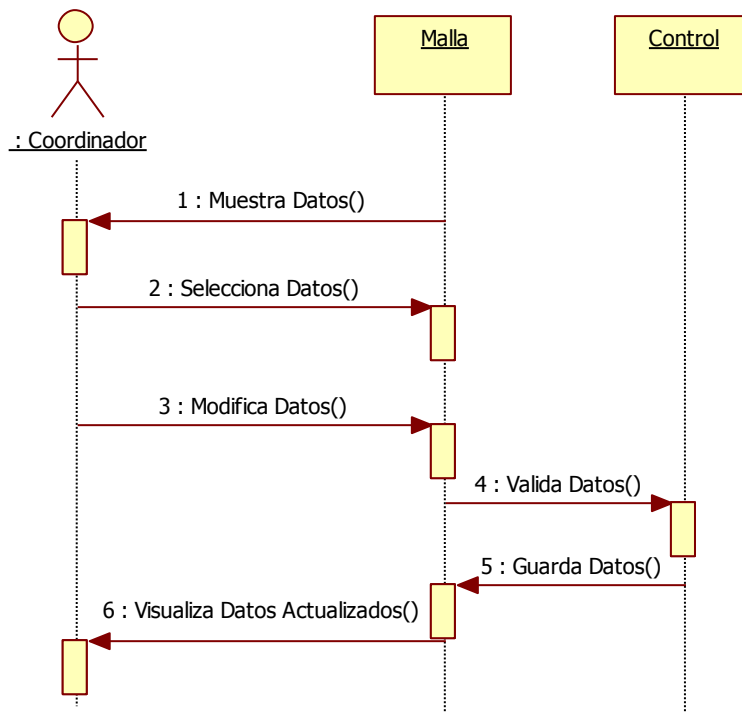


Gráfico 4.5 Diagrama de Secuencia Modificación de Malla
Fuente: StartUML

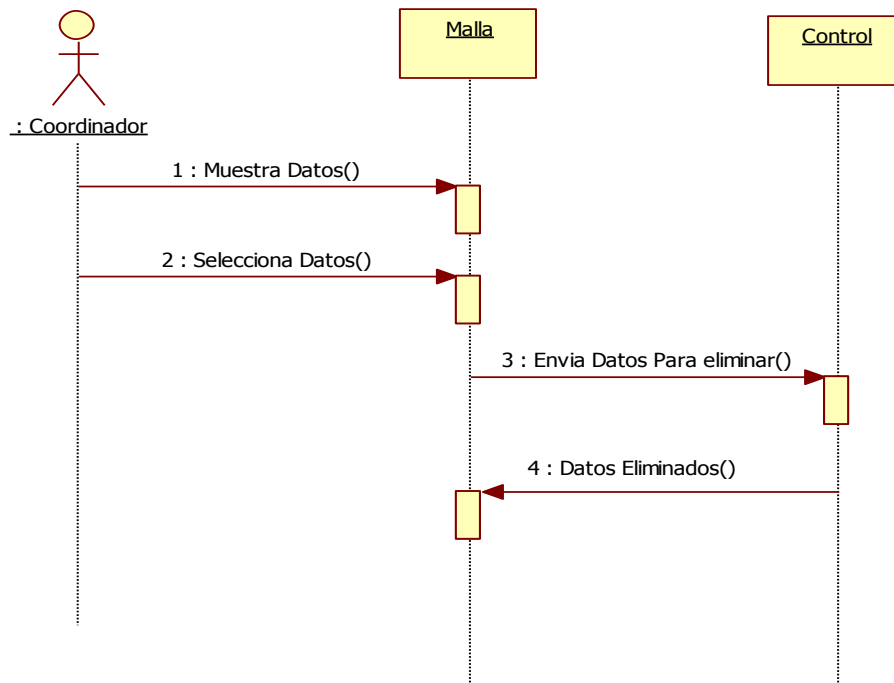


Gráfico 4.6 Diagrama de Secuencia Eliminación de Malla
Fuente: StartUML

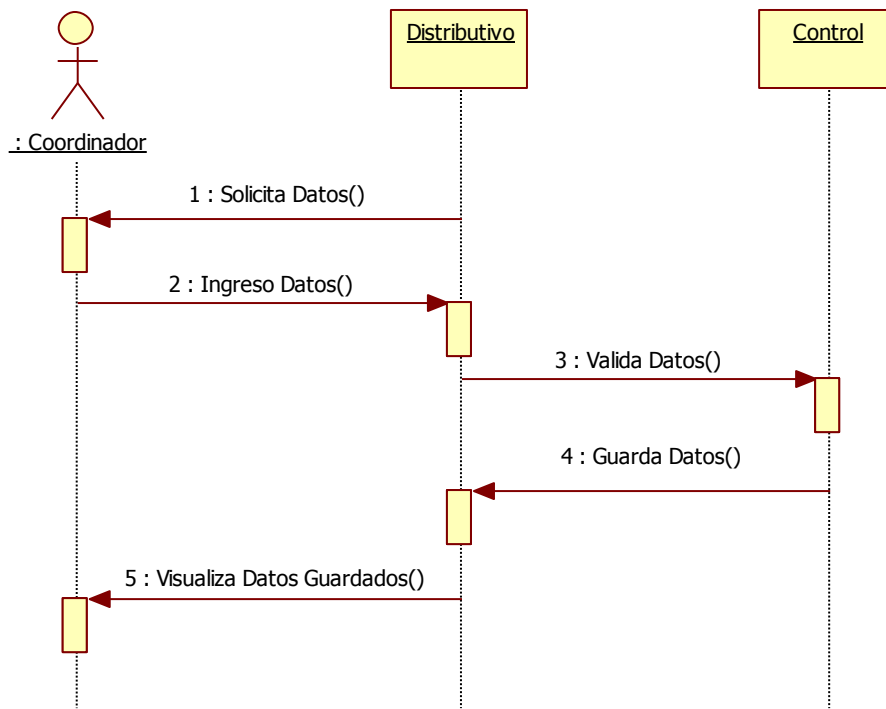


Gráfico 4.7 Diagrama de Secuencia Ingreso de Distributivo
Fuente: StartUML

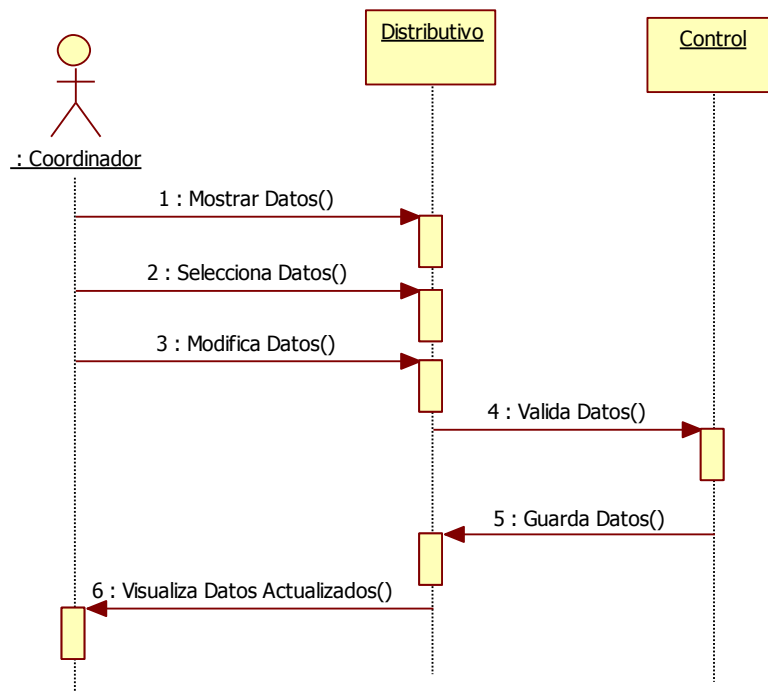


Gráfico 4.8 Diagrama de Secuencia Modificación de Distributivo
Fuente: StartUML

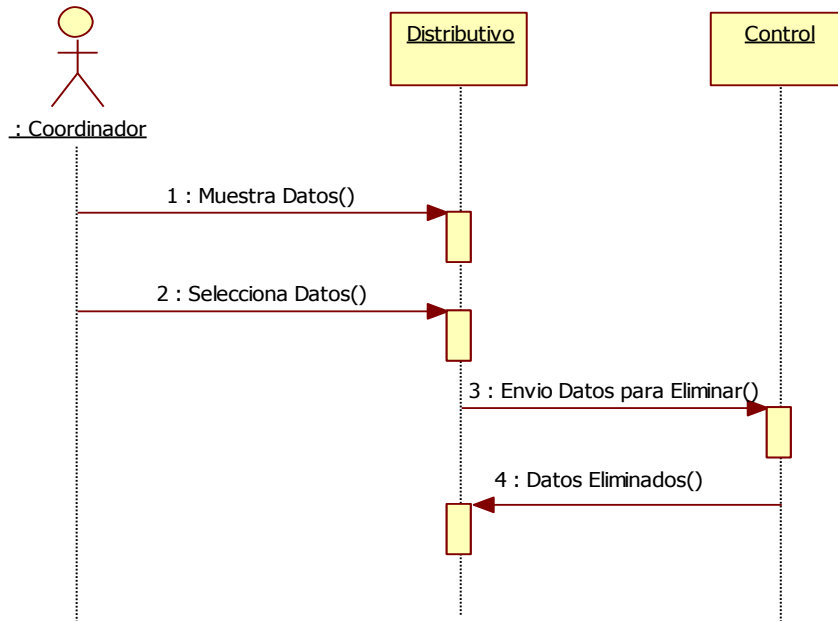


Gráfico 4.9 Diagrama de Secuencia Eliminación de Distributivo
Fuente: StartUML

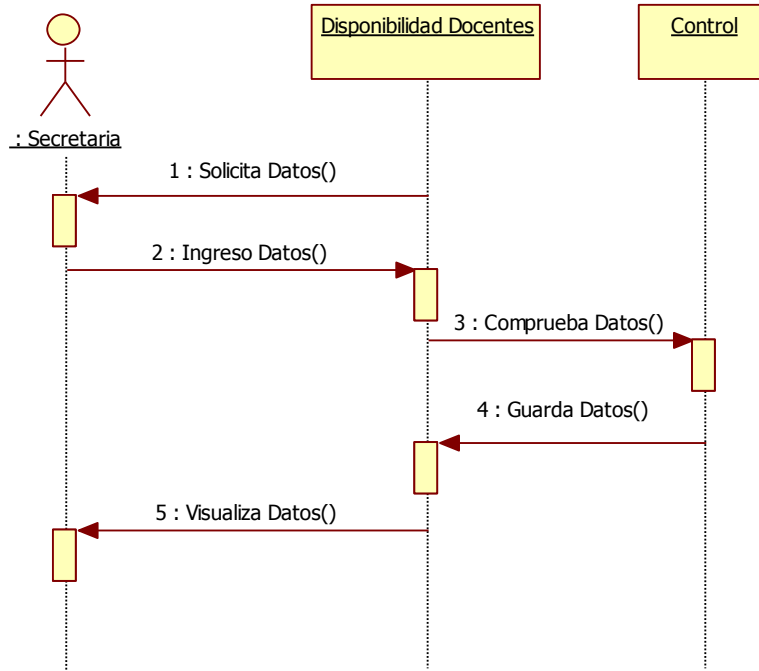


Gráfico 4.10 Diagrama de Secuencia Ingreso de Disponibilidad Docentes
Fuente: StartUML

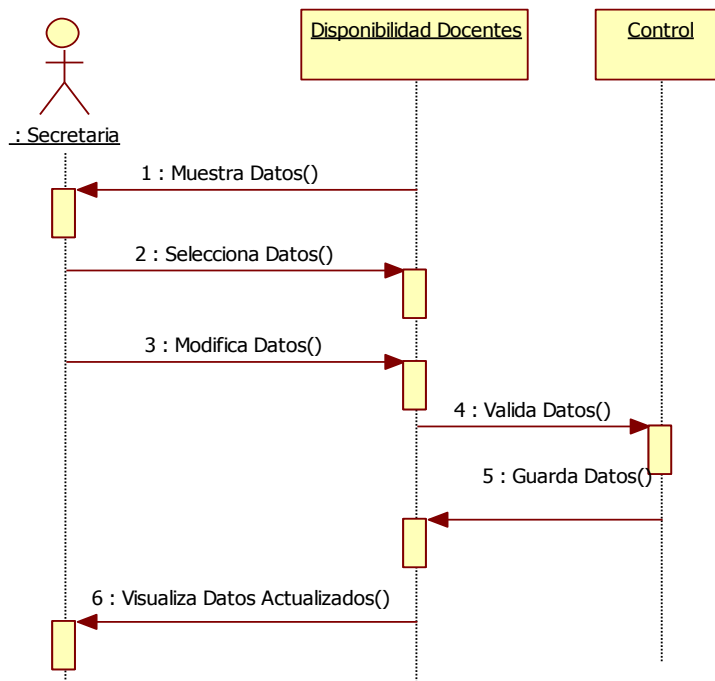


Gráfico 4.11 Diagrama de Secuencia Modificación de Disponibilidad Docentes
Fuente: StartUML

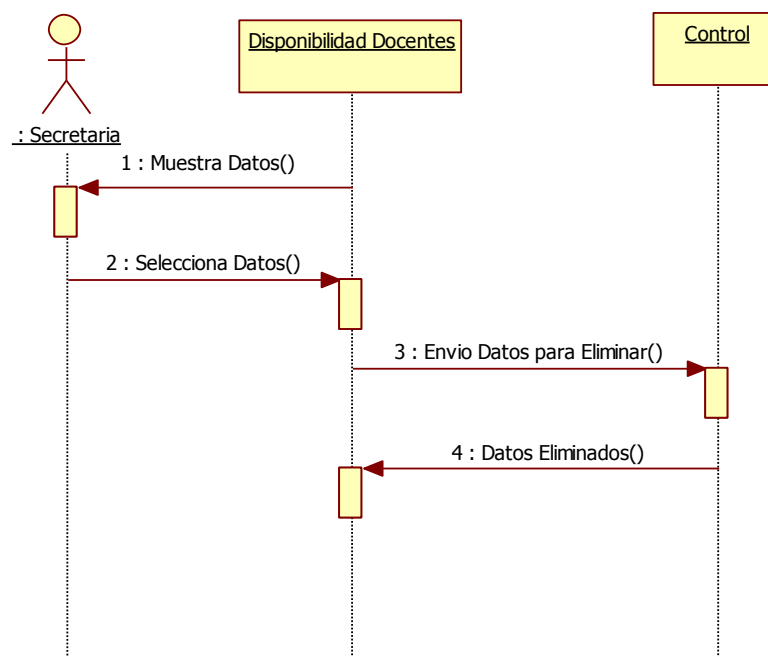


Gráfico 4.12 Diagrama de Secuencia Eliminación de Disponibilidad Docentes
Fuente: StartUML

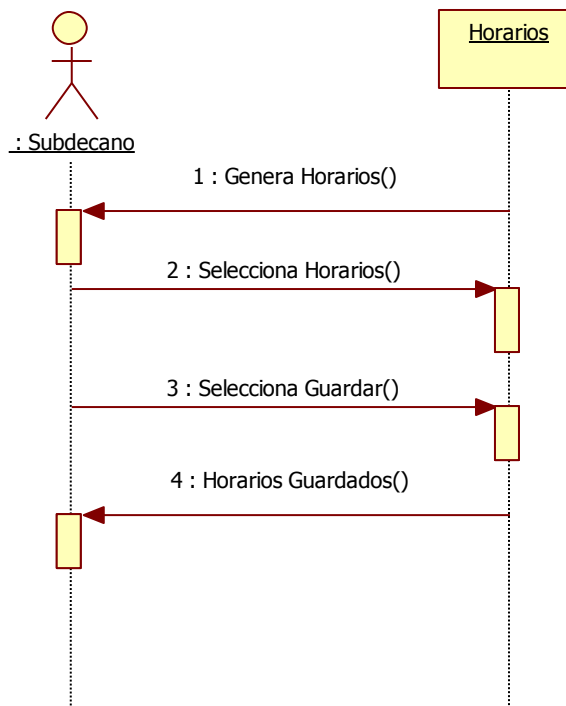


Gráfico 4.13 Diagrama de Secuencia Proceso Horarios
Fuente: StartUML

4.9 Selección del Framework o herramienta apropiada

4.9.1 Herramientas Proporcionadas

Las herramientas proporcionadas con sus respectivas licencias previas a la investigación realizada de cada una de ellas son:

- Sql Server 2008 R2
- Visual Studio 2012

4.9.2 Herramientas Soportadas

Con el análisis respetivo sobre las herramientas que se adaptan al servidor de la administración de Redes de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la UTA son:

- Asp.net
- JSP
- Postgresql
- Sql Server 2008 R2
- MySQL

4.9.3 Análisis de Base de Datos

Para el respectivo análisis se ha seleccionado 3 Gestores de Bases de Datos.

Características	SQL Server	PostgreSQL	MySQL
Sistemas Operativos	Todas las versiones de Windows	Windows, Linux, Unix, Solaris	Windows, Linux, Unix, Solaris,
Modelo de Base de Datos	DBMS relacional	DBMS relacional	DBMS relacional
Licencia	Comercial	Open Source	Open Source
Lenguaje de Implementación	C	C y C++	C
API's y métodos de Acceso	OLE DB tabular secuencia de datos (TDS) ADO.NET JDBC ODBC	Liberia Nativa C API de streaming para los grandes objetos ADO.NET JDBC ODBC	ADO.NET JDBC ODBC
Recursos Consumidos de CPU	Todos los recursos disponibles	Bastantes	Pocos
Velocidad de respuesta	Alta	Media	Alta
Lenguajes de Programación Soportados	.Net Java PHP Python Ruby Visual Basic	.Net C C++ Java Perl Python Tcl	Ada C C# C++ D Eiffel Erlang Haskell Java Objective-C OCaml Perl PHP Python Ruby Scheme Tcl
Scripts del lado del Servidor	Lenguajes Transact SQL y .NET	Funciones definidas por el usuario	Si
Métodos de Particionamiento	Tablas se pueden distribuir en varios archivos (partición horizontal); sharding través de la federación	no, pero se puede realizar utilizando la herencia de tablas	Partición horizontal, sharding con MySQL Cluster o MySQL Tela
Métodos de Replicación	Transaccional Mezcla Instantáneas	Maestro – Maestro Maestro – Esclavo Circular	Maestro – Esclavo
Esquema de Datos	Si	Si	Si
Escalabilidad	Si	Si	No
Concurrencia	Si	Si	Si
Durabilidad	Si	Si	Si
Triggers	Si	Si	Si
Seguridad	Alta	Alta	Media
Soporta Transacciones	Si	Si	No
Soporte Empresarial	Si	Si	No

Facilidad de configuración e instalación	Media	Alta	Alta
Procedimientos Almacenados	Si	Si	Si
Herramientas de Diseño Grafico	Si	Si	No

Análisis

Mediante el análisis anterior de cada gestor de base de datos se puede concluir que PostgreSQL y SQL Server tienen soporte empresarial además de contar con seguridades altas, estabilidad, confiabilidad, escalabilidad además del soporte de transacciones en cuanto a información. Sin embargo la Universidad Técnica de Ambato cuenta con las licencias y el Soporte adecuado de Microsoft lo cual pone en relieve a SQL Server.

4.9.4 Análisis de Lenguajes de Programación

Para el respectivo análisis se ha seleccionado 2 Lenguajes de Programación.

Características	ASP.NET	JSP
Plataforma	Windows, Linux	Windows, Linux, Unix
Bajo Costo	No	Si
Portabilidad	Muy Buena	Excelente
Seguridad	Excelente	Muy Buena
Estabilidad	Excelente	Excelente
Acceso a Bases de Datos	Excelente	Excelente
Multiplataforma	Buena	Excelente
Programación Orientado a Objetos	Si	Si
Bajo Requerimiento de Hardware	No	Si
Aplicaciones con Alta Complejidad	Si	Si
Fácil Desarrollo	Muy Buena	Buena
Facilidad de Ayuda	Buena	Buena
Soporte XML	Si	Si
Velocidad de Ejecución	Muy Buena	Buena
Soporte Técnico	Excelente	Muy Buena
IDEs Disponibles	Excelente	Excelente
Servidores Web disponibles en Internet	Buena	Regular
Aprendizaje	Fácil	Complejo
Framework	Si	No

Análisis

Mediante el análisis previo de cada Lenguaje de Programación se puede llegar a la Conclusión que ASP.NET tiene un aprendizaje mucho más fácil, mejor seguridad, mejor velocidad de ejecución, con un soporte técnico mucho más alto y servidores web disponibles en Internet para esta tecnología. De modo que lo coloca como la mejor opción para el desarrollo del proyecto.

4.9.5 Análisis de Métodos de Optimización

Para el respectivo análisis se seleccionó los Algoritmos Genéticos y los métodos de Optimización tradicionales.

Algoritmos Genéticos	Métodos de Optimización Tradicionales
<ul style="list-style-type: none">• Los AGs tienen descendencia múltiple, pueden explorar el espacio de soluciones en múltiples direcciones a la vez. Si un camino resulta ser un callejón sin salida, pueden eliminarlo fácilmente y continuar el trabajo en avenidas más prometedoras, dándoles una mayor probabilidad en cada ejecución de encontrar la solución.• Los AGs les permite superar incluso un enorme número de posibilidades, y encontrar con éxito resultados óptimos o muy buenos en un corto periodo de tiempo.• Los algoritmos genéticos es que se desenvuelven bien en problemas con un paisaje adaptativo complejo - aquéllos en los que la función de aptitud es discontinua, ruidosa, cambia con el tiempo, o tiene muchos óptimos locales.• Se destacan los algoritmos genéticos es su habilidad para manipular muchos parámetros simultáneamente.• Se basan en reglas probabilísticas.	<ul style="list-style-type: none">• La mayoría de los otros algoritmos son en serie y sólo pueden explorar el espacio de soluciones hacia una solución en una dirección al mismo tiempo, y si la solución que descubren resulta subóptima, no se puede hacer otra cosa que abandonar todo el trabajo hecho y empezar de nuevo.• Los métodos convencionales evalúan un cierto límite de posibilidades y se demoran mucho más tiempo llegar a una posible solución.• Muchos algoritmos de búsqueda pueden quedar atrapados en los óptimos locales: si llegan a lo alto de una colina del paisaje adaptativo, descubrirán que no existen soluciones mejores en las cercanías y concluirán que han alcanzado la mejor de todas, aunque existan picos más altos en algún otro lugar del mapa.• Siempre manipula los mismos parámetros.• Se basan en reglas determinísticas.

Análisis

Mediante el análisis previo entre los Algoritmos Genéticos y los Métodos de Optimización Tradicionales se puede llegar a la conclusión que los algoritmos genéticos

pueden explorar más soluciones en poco tiempo además de adaptarse a los medios necesarios hasta encontrar la solución. De modo que cumple las funciones necesarias para la combinación de las posibles soluciones en cuanto a la generación de horarios.

4.10 Diseño del Sistema

4.10.1 Diseño de la Base de Datos

Con el respectivo análisis de la información adquirida y los requerimientos establecidos por la Administración de Redes de la FISEI se migro la base de datos al SQL Server debido a las licencias facilitadas por la Universidad Técnica de Ambato además de la creación de tablas necesarias para el funcionamiento del Módulo.

4.10.2 Diccionario de Datos

Tablas Utilizadas por el Sistema de Control de Docentes sin el Modulo

ad_usuario	TABLA USUARIO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
adusu_codigo	int		entero largo	SI	SI	NO		Código
adusu_cedula	varchar	20	texto	SI	NO	NO		Cédula
adusu_nombre	varchar	50	texto	SI	NO	NO		Nombre
adusu_apellido	varchar	50	texto	SI	NO	NO		Apellido
adusu_titulo	varchar	50	texto	SI	NO	NO		Título
adusu_categoria	varchar	50	texto	SI	NO	NO		Categoría
adusu_telefono1	varchar	20	texto	NO	NO	NO		Teléfono1
adusu_telefono2	varchar	20	texto	NO	NO	NO		Teléfono2
adusu_telefono3	varchar	20	texto	NO	NO	NO		Teléfono3
adusu_direccion	varchar	100	texto	NO	NO	NO		Dirección
adusu_email	varchar	50	texto	NO	NO	NO		Email
adusu_foto	varchar	150	texto	NO	NO	NO		Foto
adusu_usuario	varchar	20	texto	SI	SI	NO		Usuario
adusu_contrasenia	varchar	50	texto	SI	NO	NO		Contraseña
adusu_fechaCreacion	date			SI	NO	NO		Fecha de Creación
adusu_fechaModificacion	datetime			SI	NO	NO		Fecha de Modificación
adca_estadoUsuario	int		entero largo	SI	NO	NO		Estado de Usuario
adrol_codigo	int		entero largo	SI	NO	NO		Código de Rol
adusu_tiempo	varchar	50	texto	NO	NO	NO		Tiempo
adusu_cierraregistro	bit			NO	NO	NO		Cierra Registro
adusu_admifacultad	int		entero largo	NO	NO	NO		Administrador Facultad

adusu_tituloivnivel	varchar	50	texto	NO	NO	NO		Título IV Nivel
---------------------	---------	----	-------	----	----	----	--	-----------------

Tabla 4.14 Diccionario de Tablas Usuario
Realizado por: Andrea Sotelo

universidad	TABLA UNIVERSIDAD							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
codigouni	int		entero largo	SI	SI	NO		Código
nombreuni	varchar	100	texto	NO	NO	NO		Nombre
descripuni	varchar	100	texto	NO	NO	NO		Descripción

Tabla 4.15 Diccionario de Tablas Universidad
Realizado por: Andrea Sotelo

facultad	TABLA FACULTAD							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
codigofacu	int		entero largo	SI	SI	NO		Código Facultad
codigouni	int		entero largo	SI	NO	SI	universidad	Código Universidad
nombrefacu	varchar	100	texto	NO	NO	NO		Nombre
descripcionfacu	varchar	100	texto	NO	NO	NO		Descripción
sellofacu	varchar	100	texto	NO	NO	NO		Sello

Tabla 4.16 Diccionario de Tablas Facultad
Realizado por: Andrea Sotelo

carrera	TABLA CARRERA							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
codigocarr	int		entero largo	SI	SI	NO		Código Carrera
codigofacu	int		entero largo	SI	NO	SI	facultad	Código Facultad
nombrecarr	varchar	100	texto	NO	NO	NO		Nombre
descripcioncarr	varchar	100	texto	NO	NO	NO		Descripción

Tabla 4.17 Diccionario de Tablas Carrera

Realizado por: Andrea Sotelo

profesorporcarrera	TABLA PROFESOR POR CARRERA							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
codigocarr	int		entero largo	SI	SI	NO		Código Carrera
adusu_codigo	int		entero largo	NO	NO	SI	ad_usuario	Código Usuario
tipo	varchar	50	texto	NO	NO	NO		Tipo
adusu_usuario	varchar	20	texto	NO	NO	SI	ad_usuario	Usuario

Tabla 4.18 Diccionario de Tablas Profesor por Carrera

Realizado por: Andrea Sotelo

ad_catalogo	TABLA CATALOGO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
cat_codigo	int		entero largo	SI	SI	NO		Código Catalogo
cat_tipo	int		entero largo	NO	NO	SI		Tipo
cat_descripcion	varchar	50	texto	NO	NO	NO		Descripción
adusu_codigo	int		entero largo	SI	NO	SI	ad_usuario	Código Usuario
ad_numhoras	int		entero largo	NO	NO	NO		Número de Horas
cat_ordenhoras	int		entero largo	NO	NO	NO		Orden de Horas
adusu_admifacultad	int		entero largo	NO	NO	NO		Administrador de Facultad
adusu_usuario	varchar	20	texto	NO	NO	SI	ad_usuario	Usuario

Tabla 4.19 Diccionario de Tablas Catalogo
Realizado por: Andrea Sotelo

ad_contenido	TABLA CONTENIDO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
con_codigo	int		entero largo	SI	SI	NO		Código Contenido
con_materia	varchar	150	texto	NO	NO	NO		Materia
con_tema	varchar	150	texto	NO	NO	NO		Tema
con_subtema	varchar	150	texto	NO	NO	NO		Subtema
fac_codigo	varchar	50	texto	NO	NO	NO		Código Facultad
adusu_codigo	int		entero largo	SI	NO	SI	ad_usuario	Código Usuario
adusu_usuario	varchar	20	texto	NO	NO	SI	ad_usuario	Usuario

Tabla 4.20 Diccionario de Tablas Contenido
Realizado por: Andrea Sotelo

ad_cumplimiento	TABLA CUMPLIMIENTO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
cedula	varchar	20	texto	NO	NO	NO		Cédula
adusu_nombre	varchar	50	texto	NO	NO	NO		Nombre
adusu_apellido	varchar	50	texto	NO	NO	NO		Apellido
dia_codigo	varchar	20	texto	NO	NO	NO		Código Dia
hra_hora1	time	7		NO	NO	NO		Hora1
hra_hora2	time	7		NO	NO	NO		Hora2
hra_fecharegistro	date			NO	NO	NO		Fecha de Registro
hra_reg_entrada	time	7		NO	NO	NO		Hora Entrada
hra_reg_salida	time	7		NO	NO	NO		Hora Salida
niv_codigo	varchar	3	texto	NO	NO	NO		Código Nivel
hor_paralelo	int		entero largo	NO	NO	NO		Paralelo
con_materia	varchar	150	texto	NO	NO	NO		Materia
detalleingreso	varchar	20	texto	NO	NO	NO		Detalle de Ingreso
retrasoingreso	varchar	20	texto	NO	NO	NO		Retraso de Ingreso
ingresaantes	varchar	20	texto	NO	NO	NO		Ingresas Antes
detallesalida	varchar	20	texto	NO	NO	NO		Detalle Salida
saleantes	varchar	150	texto	NO	NO	NO		Sale Antes
saletarde	varchar	150	texto	NO	NO	NO		Sale Tarde
periodo	varchar	75	texto	NO	NO	NO		Periodo

Tabla 4.21 Diccionario de Tablas Cumplimiento
Realizado por: Andrea Sotelo

ad_menu	TABLA MENÚ							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
admen_codigo	varchar	50	texto	SI	SI	NO		Código Menú
admen_descripcion	varchar	50	texto	SI	NO	NO		Descripción
admen_codigopadre	varchar	50	texto	SI	NO	NO		Código Padre
admen_codigonivel	int		entero largo	SI	NO	NO		Código Nivel
admen_link	varchar	150	texto	NO	NO	NO		Link
admen_imagen	varchar	150	texto	NO	NO	NO		Imagen
addca_estadomenu	int		entero largo	SI	NO	NO		Estado de Menú
admen_nivel	varchar	20	texto	NO	NO	NO		Nivel
admen_mostrar	varchar	2	texto	NO	NO	NO		Mostrar
admen_hijo	int		entero largo	NO	NO	NO		Hijo
admenu_orden	int		entero largo	NO	NO	NO		Orden

Tabla 4.22 Diccionario de Tablas Menú
Realizado por: Andrea Sotelo

ad_rol	TABLA ROL							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
adrol_codigo	int		entero largo	SI	SI	NO		Código Rol
adrol_descripcion	varchar	50	texto	SI	NO	NO		Descripción
addca_estadorol	varchar	50	texto	SI	NO	NO		Estado del Rol

Tabla 4.23 Diccionario de Tablas Rol
Realizado por: Andrea Sotelo

ad_menurol	TABLA MENÚ ROL							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
admrl_codigo	int		entero largo	SI	SI	SI		Código Menú Rol
admen_codigo	varchar	50	texto	SI	SI	SI	ad_menu	Código Menú
adrol_codigo	int		entero largo	SI	SI	SI	ad_rol	Código Rol
addca_estadorol	varchar	50	texto	SI	NO	NO		Estado del Rol

Tabla 4.24 Diccionario de Tablas Menú Rol

Realizado por: Andrea Sotelo

ad_niveles	TABLA NIVELES							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
niv_secuencial	int		entero largo	SI	SI	NO		Secuencia
niv_codigo	int		entero largo	SI	SI	NO		Código Niveles
niv_paralelo	varchar	50	texto	SI	SI	NO		Paralelo
niv_grupo	int		entero largo	SI	SI	NO		Grupo
fac_codigo	varchar	50	texto	SI	SI	NO		Código facultad
adusu_admifacultad	int		entero largo	NO	NO	NO		Adminsitrador Facultad

Tabla 4.25 Diccionario de Tablas Niveles

Realizado por: Andrea Sotelo

cronograma	TABLA CRONOGRAMA							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
codigocron	int		entero largo	SI	SI	NO		Código Cronograma
nombrecron	varchar	50	texto	NO	NO	NO		Nombre
fechainicron	date			NO	NO	NO		Fecha de Inicio
fechafincron	date			NO	NO	NO		Fecha de Fin
estadocron	bit			NO	NO	NO		Estado

Tabla 4.26 Diccionario de Tablas Cronograma
Realizado por: Andrea Sotelo

subcronograma	TABLA SUBCRONOGRAMA							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
codigosubcron	int		entero largo	SI	SI	NO		Código Subcronograma
fechainicio	varchar	50	texto	NO	NO	NO		Fecha de Inicio
fechafinal	date			NO	NO	NO		Fecha de Fin
nombre	date			NO	NO	NO		Nombre
estadocron	bit			NO	NO	NO		Estado
codigocron	int		entero largo	NO	NO	NO		Código Cronograma
adusu_admifacultad	int		entero largo	NO	NO	NO		Adminsitrador Facultad

Tabla 4.27 Diccionario de Tablas Subcronograma
Realizado por: Andrea Sotelo

detallesubcrono	TABLA DETALLE DEL SUBCRONOGRAMA							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
codigosubcron	int		entero largo	NO	NO	SI		Código Subcronograma
fechaini	date			NO	NO	NO		Fecha de Inicio
fechafin	date			NO	NO	NO		Fecha de Fin
nombre	varchar	50	texto	NO	NO	NO		Nombre
estado	bit			NO	NO	NO		Estado
adusu_codigo	int		entero largo	NO	NO	SI	ad_usuario	Código Usuario
adusu_usuario	varchar	20	texto	NO	NO	SI	ad_usuario	Usuario

Tabla 4.28 Diccionario de Tablas Detalle del Subcronograma

Realizado por: Andrea Sotelo

ad_horario_temporal	TABLA HORARIO TEMPORAL							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
hra_codigo	int		entero largo	SI	SI	NO		Código
niv_codigo	int		entero largo	SI	NO	SI	ad_niveles	Código Niveles
hor_paralelo	varchar	3	texto	SI	NO	SI	ad_niveles	Paralelo
hor_grupo	int		entero largo	NO	NO	SI	ad_niveles	Grupo
dia_codigo	varchar	20	texto	SI	SI	NO		Código Día
hra_hora1	time	7		SI	SI	NO		Hora1
hra_hora2	time	7		NO	NO	NO		Hora2
mat_codigo	varchar	50	texto	SI	SI	NO		Código Materia
adusu_codigo	int		entero largo	SI	SI	SI	ad_usuario	Código Usuario
adusu_usuario	varchar	20	texto	SI	NO	SI	ad_usuario	Usuario
fac_codigo	varchar	7	texto	SI	SI	SI		Código Facultad
hra_tipohoras	varchar	50	texto	NO	NO	NO		Tipo de Horas
hra_estado	int		entero largo	NO	NO	NO		Estado
tipo	varchar	50	texto	NO	NO	NO		Tipo
codigocron	int		entero largo	SI	NO	SI	cronograma	Código Cronograma
adusu_admifacultad	int		entero largo	NO	NO	NO		Administrador Facultad

Tabla 4.29 Diccionario de Tablas Horario Temporal

Realizado por: Andrea Sotelo

ad_recuperaciones	TABLA RECUPERACIONES							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
reg_codigo	int		entero largo	SI	SI	NO		Código Recuperaciones
adusu_cedula	varchar	20	texto	SI	NO	NO		Cédula
adusu_nombre	varchar	50	texto	SI	NO	NO		Nombre
adusu_apellido	varchar	50	texto	SI	NO	NO		Apellido
dia_codigo	varchar	20	texto	SI	NO	NO		Código Día
hra_hora1	time	7		SI	NO	NO		Hora1
hra_hora2	time	7		SI	NO	NO		Hora2
hra_tipohoras	varchar	50	texto	SI	NO	NO		Tipo de Horas
hra_fecharegistro	date		entero largo	NO	NO	NO		Fecha de Registro
hra_reg_entrada	time	7		NO	NO	NO		Hora Entrada
hra_reg_salida	time	7		NO	NO	NO		Hora Salida
estado_registro	varchar	50	texto	NO	NO	NO		Estado de Registro
niv_codigo	int		entero largo	SI	NO	NO		Código Niveles
hor_paralelo	varchar	3	texto	SI	NO	NO		Paralelo
hor_grupo	int		entero largo	NO	NO	NO		Grupo
con_materia	varchar	150	texto	NO	NO	NO		Materia
con_tema	varchar	150	texto	NO	NO	NO		Tema
con_subtema	varchar	150	texto	NO	NO	NO		Subtema
reg_observacion	varchar	200	texto	NO	NO	NO		Observación de Registro
reg_anexo	varchar	200	texto	NO	NO	NO		Anexo
fac_codigo	varchar	20	texto	SI	NO	NO		Código Registro
hra_hora1_new	time	7		NO	NO	NO		Nueva Hora1
hra_hora2_new	time	7		NO	NO	NO		Nueva Hora2
hra_fecharegistro_new	date			NO	NO	NO		Nueva Fecha de Registro

estado_registro_new	varchar	50	texto	NO	NO	NO		Nuevo Estado de Registro
observaciones	varchar	250	texto	NO	NO	NO		Observaciones

Tabla 4.30 Diccionario de Tablas Recuperaciones

Realizado por: Andrea Sotelo

ad_registro	TABLA REGISTRO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
reg_codigo	int		entero largo	SI	SI	NO		Código Recuperaciones
adusu_cedula	varchar	20	texto	SI	NO	NO		Cédula
adusu_nombre	Varchar	50	texto	SI	NO	NO		Nombre
adusu_apellido	Varchar	50	texto	SI	NO	NO		Apellido
dia_codigo	Varchar	20	texto	SI	NO	NO		Código Día
hra_hora1	time	7		SI	NO	NO		Hora1
hra_hora2	time	7		SI	NO	NO		Hora2
hra_tipohoras	varchar	50	texto	SI	NO	NO		Tipo de Horas
hra_fecharegistro	date			NO	NO	NO		Fecha de Registro
hra_reg_entrada	time	7		NO	NO	NO		Hora Entrada
hra_reg_salida	time	7		NO	NO	NO		Hora Salida
estado_registro	varchar	50	texto	NO	NO	NO		Estado de Registro
niv_codigo	int		entero largo	SI	NO	NO		Código Niveles
hor_paralelo	varchar	3	texto	SI	NO	NO		Paralelo
hor_grupo	int		entero largo	NO	NO	NO		Grupo
con_materia	varchar	150	texto	NO	NO	NO		Materia
con_tema	varchar	150	texto	NO	NO	NO		Tema
con_subtema	varchar	150	texto	NO	NO	NO		Subtema

reg_observacion	varchar	200	texto	NO	NO	NO		Observación de Registro
reg_anexo	varchar	200	texto	NO	NO	NO		Anexo
fac_codigo	varchar	20	texto	SI	NO	NO		Código Registro

Tabla 4.31 Diccionario de Tablas Registro

Realizado por: Andrea Sotelo

creacionhorario	TABLA CREACIÓN DE HORARIO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
periodo	varchar	50	texto	NO	NO	NO		Periodo
cedula	varchar	10	texto	NO	NO	NO		Cédula
nombre	varchar	50	texto	NO	NO	NO		Nombre
apellido	varchar	50	texto	NO	NO	NO		Apellido
hra_hora1	time	7		NO	NO	NO		Hora1
hra_hora2	time	7		NO	NO	NO		Hora2
lunes	varchar	50	texto	NO	NO	NO		Lunes
mlunes	varchar	50	texto	NO	NO	NO		MLunes
martes	varchar	50	texto	NO	NO	NO		Maertes
mmartes	varchar	50	texto	NO	NO	NO		MMartes
miercoles	varchar	50	texto	NO	NO	NO		Miercoles
mmiercoles	varchar	50	texto	NO	NO	NO		MMiercoles
jueves	varchar	50	texto	NO	NO	NO		Jueves
mjueves	varchar	50	texto	NO	NO	NO		MJueves
viernes	varchar	50	texto	NO	NO	NO		Viernes
mviernes	varchar	50	texto	NO	NO	NO		MViernes
sabado	varchar	50	texto	NO	NO	NO		Sábado

msabado	varchar	50	texto	NO	NO	NO		MSábado
domingo	varchar	50	texto	NO	NO	NO		Domingo
mdomingo	varchar	50	texto	NO	NO	NO		MDomingo

Tabla 4.32 Diccionario de Tablas Creación de Horario
Realizado por: Andrea Sotelo

creacionhorarionivel	TABLA CREACIÓN DE HORARIO NIVEL							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
periodo	varchar	50	texto	NO	NO	NO		Periodo
hra_hora1	time	7		NO	NO	NO		Hora1
hra_hora2	time	7		NO	NO	NO		Hora2
lunes	varchar	50	texto	NO	NO	NO		Lunes
mlunes	varchar	50	texto	NO	NO	NO		MLunes
martes	varchar	50	texto	NO	NO	NO		Maertes
mmartes	varchar	50	texto	NO	NO	NO		MMartes
miercoles	varchar	50	texto	NO	NO	NO		Miercoles
mmiercoles	varchar	50	texto	NO	NO	NO		MMiercoles
jueves	varchar	50	texto	NO	NO	NO		Jueves
mjueves	varchar	50	texto	NO	NO	NO		MJueves
viernes	varchar	50	texto	NO	NO	NO		Viernes
mviernes	varchar	50	texto	NO	NO	NO		MViernes
sabado	varchar	50	texto	NO	NO	NO		Sábado
msabado	varchar	50	texto	NO	NO	NO		MSábado
domingo	varchar	50	texto	NO	NO	NO		Domingo

mdomingo	varchar	50	texto	NO	NO	NO		MDomingo
----------	---------	----	-------	----	----	----	--	----------

Tabla 4.33 Diccionario de Tablas Creación de Horario Nivel
Realizado por: Andrea Sotelo

res_horario	TABLA RES HORARIO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
hra_codigo	int		entero largo	NO	NO	NO		Código
niv_codigo	int		entero largo	NO	NO	NO		Código Niveles
hor_paralelo	varchar	3	texto	NO	NO	NO		Paralelo
hor_grupo	int		entero largo	NO	NO	NO		Grupo
dia_codigo	varchar	20	texto	NO	NO	NO		Código Día
hra_hora1	time	7		NO	NO	NO		Hora1
hra_hora2	time	7		NO	NO	NO		Hora2
mat_codigo	varchar	50	texto	NO	NO	NO		Código Materia
adusu_codigo	int		entero largo	NO	NO	NO		Código Usuario
adusu_usuario	varchar	20	texto	NO	NO	NO		Usuario
fac_codigo	varchar	7	texto	NO	NO	NO		Código Facultad
hra_tipohoras	varchar	50	texto	NO	NO	NO		Tipo de Horas
hra_estado	int		entero largo	NO	NO	NO		Estado
tipo	varchar	50	texto	NO	NO	NO		Tipo
codigocron	int		entero largo	NO	NO	NO		Código Cronograma

Tabla 4.34 Diccionario de Tablas Res Horario
Realizado por: Andrea Sotelo

ad_registrotemporal	TABLA REGISTRO TEMPORAL							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
reg_codigo	int		entero largo	SI	SI	NO		Código Registro
hra_codigo	int		entero largo	NO	NO	NO		CódigoHora
hra_reg_entrada	time	7		NO	NO	NO		Hora Entrada
hra_reg_salida	time	7		NO	NO	NO		Hora Salida
estado_registro	varchar	50	texto	NO	NO	NO		Estado de Registro
con_codigo	int		entero largo	NO	NO	NO		Código Contenido
hra_fecharegistro	date			NO	NO	NO		Fecha de Registro

Tabla 4.35 Diccionario de Tablas Registro Temporal
Realizado por: Andrea Sotelo

ad_registrotemporal_amw	TABLA REGISTRO TEMPORAL AMW							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
reg_codigo	int		entero largo	SI	SI	NO		Código Registro
hra_codigo	int		entero largo	NO	NO	NO		CódigoHora
hra_reg_entrada	time	7		NO	NO	NO		Hora Entrada
hra_reg_salida	time	7		NO	NO	NO		Hora Salida
estado_registro	varchar	50	texto	NO	NO	NO		Estado de Registro
con_codigo	int		entero largo	NO	NO	NO		Código Contenido
hra_fecharegistro	date			NO	NO	NO		Fecha de Registro

Tabla 4.36 Diccionario de Tablas Registro Temporal AMW
Realizado por: Andrea Sotelo

tb_huellas	TABLA HUELLAS							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
cod_huella	int		entero largo	SI	SI	NO		Código Huella
cod_cedula	varchar	14	texto	SI	NO	NO		Código Cedula
huella1	varchar	2000	texto	NO	NO	NO		Huella1
descripcion	varchar	30	texto	NO	NO	NO		Descripción

Tabla 4.37 Diccionario de Tablas Huellas

Realizado por: Andrea Sotelo

tipo	TABLA TIPO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
nombre	varchar	50	texto	SI	SI	NO		Nombre
descripcion	varchar	50	texto	NO	NO	NO		Descripción

Tabla 4.38 Diccionario de Tablas Tipo

Realizado por: Andrea Sotelo

validarcedula	TABLA VALIDAR CEDULA							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
count	int		entero largo	NO	NO	NO		Contar

Tabla 4.39 Diccionario de Tablas Validar Cédula

Realizado por: Andrea Sotelo

validarlogin	TABLA VALIDAR LOGIN							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
count	int		entero largo	NO	NO	NO		Contar

Tabla 4.40 Diccionario de Tablas Validar Login
Realizado por: Andrea Sotelo

Tablas Utilizadas por el Módulo de Control de Distributivos y Generación de Horarios

➤ Dentro del Módulo también se utiliza la Tabla Usuario y la Tabla Cronograma

validarlogin	TABLA AULAS							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
id	int		entero largo	SI	SI	NO		Id
Nombre	varchar	max	texto	NO	NO	NO		Nombre
Tipo	varchar	200	texto	NO	NO	NO		Tipo
cupo	int		entero largo	NO	NO	NO		Cupo
piso	varchar	200	texto	NO	NO	NO		Piso
ubicacion	varchar	max	texto	NO	NO	NO		Ubicación
habilitado	bit			NO	NO	NO		Habilitado
codigofacu	int		entero largo	NO	NO	SI	facultad	Código Facultad

Tabla 4.41 Diccionario de Tablas Aulas
Realizado por: Andrea Sotelo

jornadas		TABLA JORNADAS						
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
id_jornada	int		entero largo	SI	SI	NO		Id
jornada	varchar	50	texto	NO	NO	NO		Jornada

Tabla 4.42 Diccionario de Tablas Jornadas
Realizado por: Andrea Sotelo

materias		TABLA MATERIAS						
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
Id_materia	int		entero largo	SI	SI	NO		Id
materia	varchar	100	texto	NO	NO	NO		Materia
facultad	int		entero largo	SI	NO	SI	facultad	Facultad

Tabla 4.43 Diccionario de Tablas Materias
Realizado por: Andrea Sotelo

niveles		TABLA NIVELES						
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
id	int		entero largo	SI	SI	NO		Id
niv_nombre	varchar	10	texto	NO	NO	NO		Nombre
niv_descripcion	varchar	100	texto	NO	NO	NO		Descripción
jornada	int		entero largo	SI	NO	SI	jornada	Jornada

Tabla 4.44 Diccionario de Tablas Niveles
Realizado por: Andrea Sotelo

mall	TABLA MALLA							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
id	int		entero largo	SI	SI	NO		Id
m_tipo	varchar	50	texto	SI	NO	NO		Tipo
m_materia	int		entero largo	NO	NO	NO		Materia
carrera	int		entero largo	NO	NO	NO		Carrera
nivel	int		entero largo	NO	NO	NO		Nivel
periodo	int		entero largo	NO	NO	NO		Periodo
creditos	int		entero largo	NO	NO	NO		Hora

Tabla 4.45 Diccionario de Tablas Malla

Realizado por: Andrea Sotelo

distributivo	TABLA DISTRIBUTIVO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
id	int		entero largo	SI	SI	NO		Id
adusu_codigo	int		entero largo	SI	SI	SI	ad_usuario	Código Usuario
adusu_usuario	varchar	20	texto	SI	SI	SI	ad_usuario	Usuario
periodo	int		entero largo	SI	SI	SI	cronograma	Periodo
malla	int		entero largo	SI	SI	SI	malla	Malla
paralelo	varchar	4	texto	SI	SI	NO		Paralelo
cupo	int		entero largo	SI	NO	NO		Cupo
observacion	varchar	300	texto	NO	NO	NO		Observación
detallereforma	varchar	700	texto	NO	NO	NO		Detalle de la Reforma
estado	bit			SI	NO	NO		Estado

Tabla 4.46 Diccionario de Tablas Distributivo

Realizado por: Andrea Sotelo

Distributivos_HAC		TABLA DISTRIBUTIVO HAC						
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
id	int		entero largo	SI	SI	NO		Id
adusu_codigo	int		entero largo	SI	SI	SI	ad_usuario	Código Usuario
adusu_usuario	varchar	20	texto	SI	SI	SI	ad_usuario	Usuario
periodo	int		entero largo	SI	SI	SI	cronograma	Periodo
descripcion	varchar	MAX	texto	SI	NO	NO		Descripción
horas	int		entero largo	SI	NO	NO		Horas
cupo	int		entero largo	SI	NO	NO		Cupo
observacion	varchar	MAX	texto	NO	NO	NO		Observación
detallereforma	varchar	MAX	texto	NO	NO	NO		Detalle de la Reforma
estado	bit			SI	NO	NO		Estado

Tabla 4.47 Diccionario de Tablas Distributivo HAC

Realizado por: Andrea Sotelo

disponibilidad	TABLA DISPONIBILIDAD							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
id	int		entero largo	SI	SI	NO		Id
adusu_codigo	int		entero largo	SI	SI	SI	ad_usuario	Código Usuario
adusu_usuario	varchar	20	texto	NO	NO	SI	ad_usuario	Usuario
periodo	int		entero largo	NO	NO	SI	cronograma	Periodo
HoraInicio	time	7		NO	NO	NO		Hora Inicio
HoraFin	time	7		NO	NO	NO		Hora Fin
Jornada	varchar	20	texto	SI	NO	NO		Jornada
Dia	varchar	50	texto	NO	NO	NO		Día

Tabla 4.48 Diccionario de Tablas Disponibilidad

Realizado por: Andrea Sotelo

horarios	TABLA HORARIOS							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
id_horario	int		entero largo	SI	SI	NO		Id
h_inicio	varchar	50	texto	SI	NO	NO		Hora Inicio
h_fin	varchar	50	texto	SI	NO	NO		Hora Fin
jornada	varchar	50	texto	NO	NO	SI	jornadas	Jornada

Tabla 4.49 Diccionario de Tablas Horarios

Realizado por: Andrea Sotelo

horario	TABLA HORARIO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
Id	int		entero largo	SI	SI	NO		Id
hora	varchar	20	texto	NO	NO	NO		Hora1
lunes	bit			NO	NO	NO		Lunes
martes	bit			NO	NO	NO		Martes
miercoles	bit			NO	NO	NO		Miércoles
jueves	bit			NO	NO	NO		Jueves
viernes	bit			NO	NO	NO		Viernes
sabado	bit			NO	NO	NO		Sábado
domingo	bit			NO	NO	NO		Domingo
disponibilidad	bit			NO	NO	NO		MDomingo

Tabla 4.50 Diccionario de Tablas Distributivo HAC

Realizado por: Andrea Sotelo

ad_horario	TABLA HORARIO GENERAL							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
hra_codigo	int		entero largo	SI	SI	NO		Código
niv_codigo	int		entero largo	SI	NO	SI	ad_niveles	Código Niveles
hor_paralelo	varchar	3	texto	SI	NO	SI	ad_niveles	Paralelo
hor_grupo	int		entero largo	NO	NO	SI	ad_niveles	Grupo
dia_codigo	varchar	20	texto	SI	SI	NO		Código Día
hra_hora1	time	7		SI	SI	NO		Hora1
hra_hora2	time	7		NO	NO	NO		Hora2
mat_codigo	varchar	50	texto	SI	SI	NO		Código Materia
adusu_codigo	int		entero largo	SI	SI	SI	ad_usuario	Código Usuario
adusu_usuario	varchar	20	texto	SI	NO	SI	ad_usuario	Usuario
fac_codigo	varchar	7	texto	SI	SI	SI		Código Facultad
hra_tipohoras	varchar	50	texto	NO	NO	NO		Tipo de Horas
hra_estado	int		entero largo	NO	NO	NO		Estado
tipo	varchar	50	texto	NO	NO	NO		Tipo
codigocron	int		entero largo	SI	NO	SI	cronograma	Código Cronograma
adusu_admifacultad	int		entero largo	NO	NO	NO		Administrador Facultad
codigosubcron	int		entero largo	NO	NO	NO	subcronograma	Código subcronograma

Tabla 4.51 Diccionario de Tablas Horario General

Realizado por: Andrea Sotelo

login_administrador	TABLA LOGIN ADMINISTRADOR							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
Id	int		entero largo	SI	SI	NO		Id
Codigo	varchar	50	texto	NO	NO	NO		Código
Nombre	varchar	200	texto	NO	NO	NO		Nombre
Usuario	varchar	200	texto	NO	NO	NO		Usuario
Rol	varchar	200	texto	NO	NO	NO		Rol
Contraseña	varchar	MAX	texto	NO	NO	NO		Contraseña
repita_password	varchar	MAX	texto	NO	NO	NO		Repita Contraseña
Correo	varchar	200	texto	NO	NO	NO		Correo
telefono	nchar	10	texto	NO	NO	NO		Teléfono
celular	nchar	10	texto	NO	NO	NO		Celular

Tabla 4.52 Diccionario de Tablas Login Administrador

Realizado por: Andrea Sotelo

login_horarios_distributivos	TABLA LOGIN HORARIOS DISTRIBUTIVO							
NOMBRE	TIPO DE DATO	TAMAÑO	FORMATO	REQ.	PK	FK	TABLA PADRE	DESCRIPCIÓN
Id	int		entero largo	SI	SI	NO		Id
Nombre	varchar	200	texto	NO	NO	NO		Nombre
Pass	varchar	MAX	texto	NO	NO	NO		Contraseña

Tabla 4.53 Diccionario de Tablas Login Horarios Distributivo

Realizado por: Andrea Sotelo

4.11 Diseño de la Interfaz

El Diseño de Interfaz se enfoca en la creación de roles para poder acceder a las diferentes administraciones que se enfoca el Sistema que además está acorde a los requerimientos establecidos por la Administración de Redes. El Sistema tendrá administración de Usuarios.

Pantalla de Ingreso al Módulo

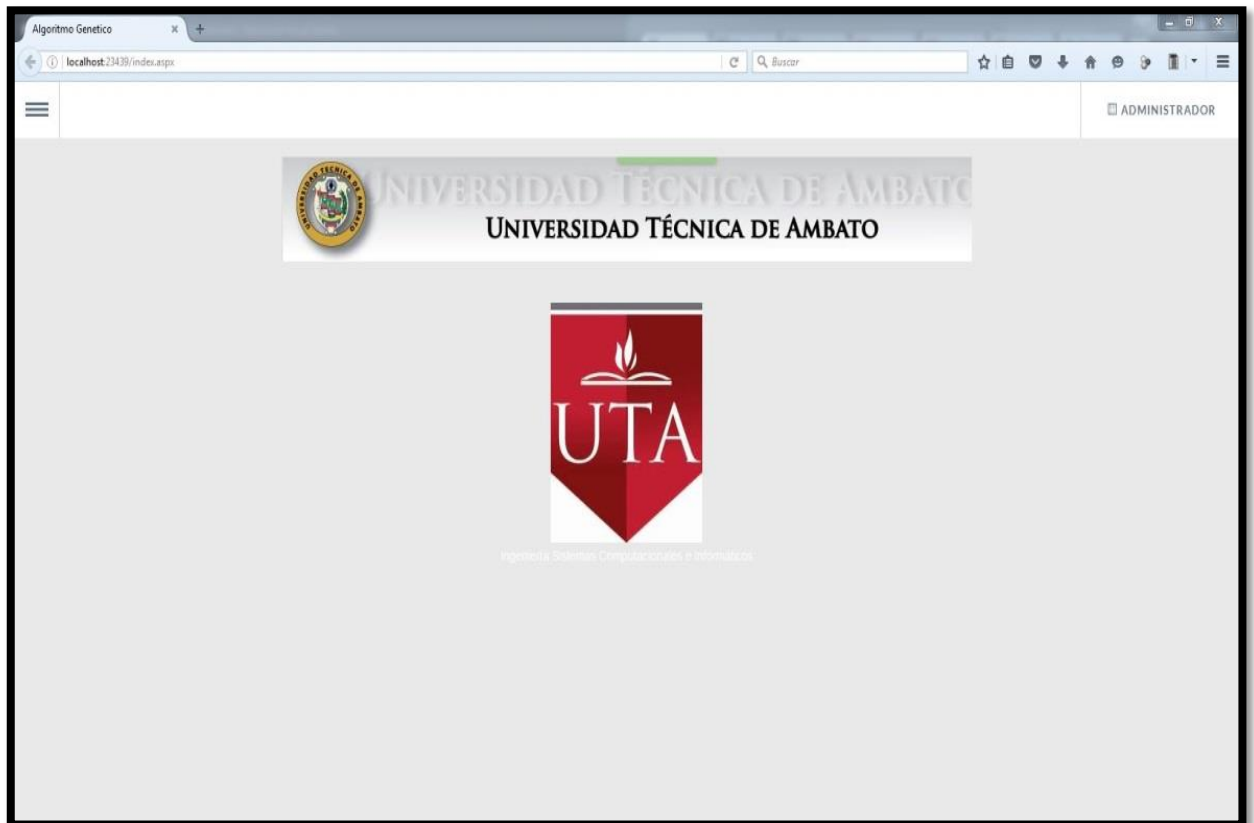


Gráfico 4.14 Diseño de Interfaz Página Inicio
Elaborado por: Andrea Sotelo

Esta pantalla es la que aparecerá para todos los usuarios cuando el módulo se ejecutado.

1. Ingreso General

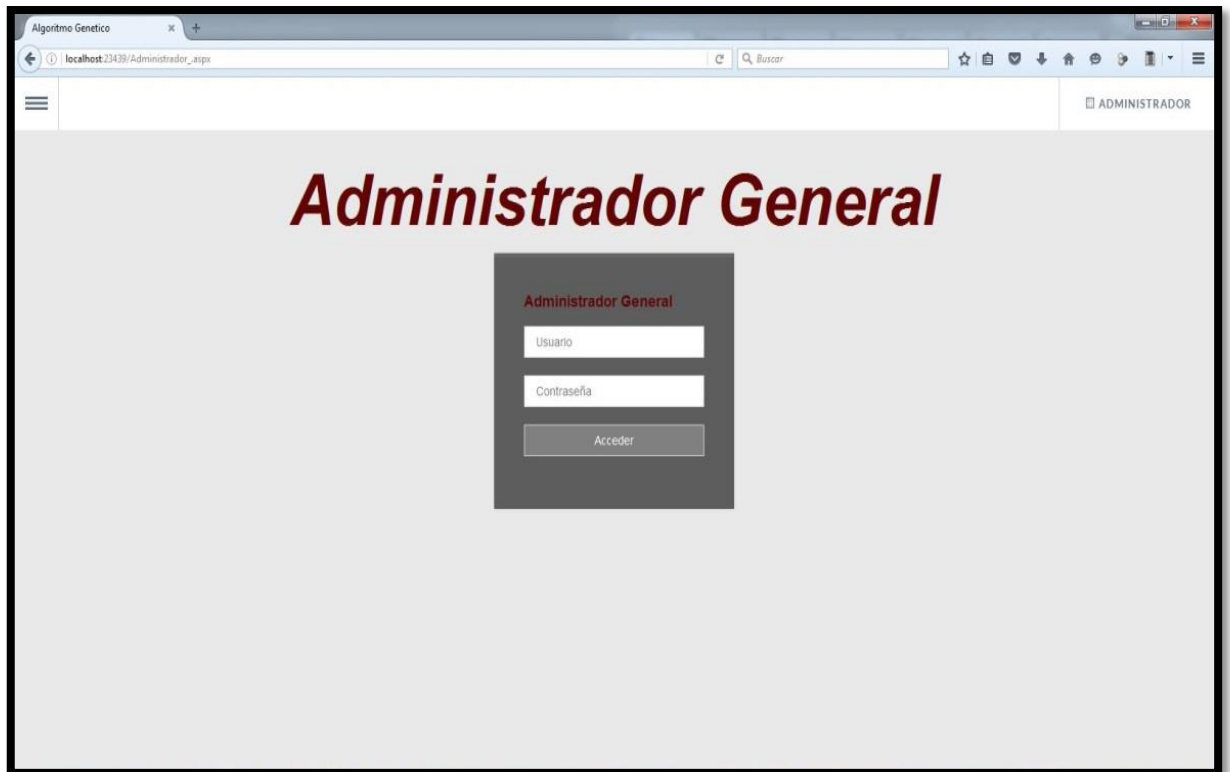


Gráfico 4.15 Diseño de Interfaz Página Ingreso
Elaborado por: Andrea Sotelo

Esta pantalla se ingresará Usuario y Contraseña para poder ingresar a las administraciones existentes en el módulo.

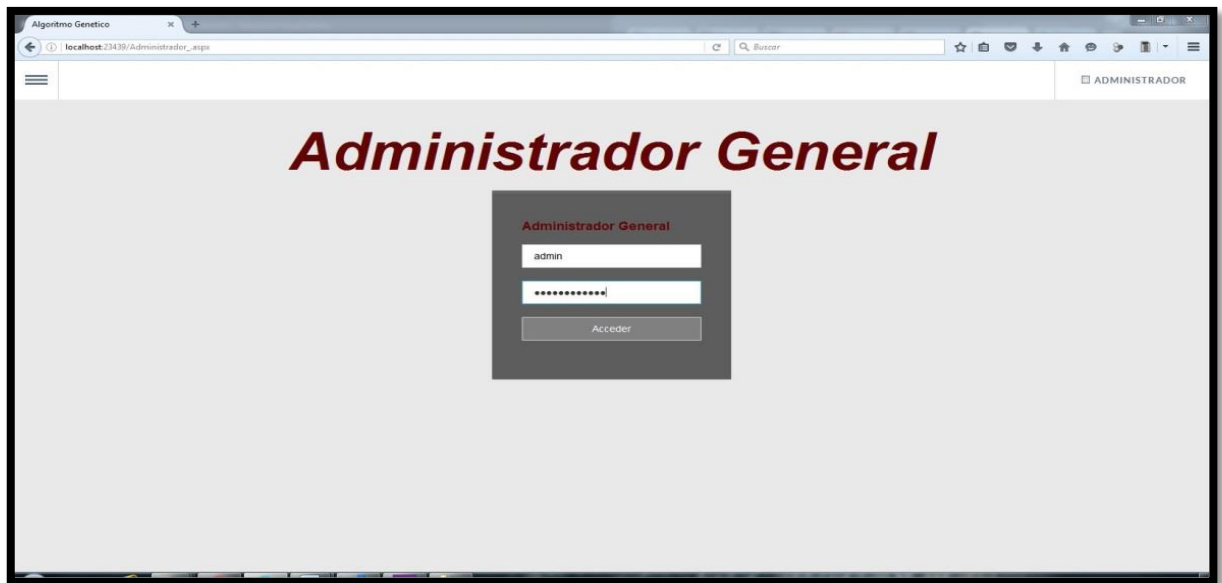


Gráfico 4.16 Diseño de Interfaz Página Ingreso
Elaborado por: Andrea Sotelo

Cuando se haya Ingresado el Usuario y Contraseña previamente asignado se podrá acceder a las administraciones.

2. Pantalla de Administraciones

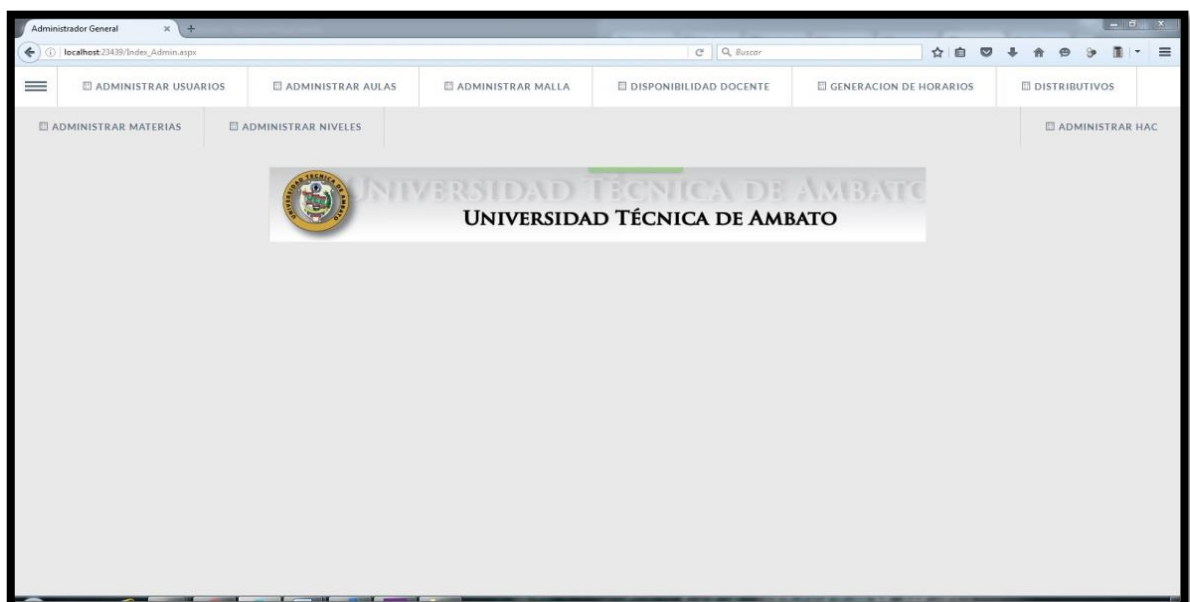


Gráfico 4.17 Diseño de Interfaz Página Principal
Elaborado por: Andrea Sotelo

Esta pantalla muestra las diferentes administraciones que maneja el Módulo como: Administrar Usuarios, Administrar Aulas, Administrar Malla, Disponibilidad Docente, Generación de Horarios.

2.1. Administrar Usuarios

Codigo	Nombre	Usuario	Rol	contraseña	repita_password	correo	telefono	celular	id	Seleccionar
001	Andres	admin	Administrador	andreasotelo	andreasotelo	andyviera212@hotmail.com	032829730	0995351475	2	
005	Prueba	prueba	Docente	1234567	1234567	andyviera212@hotmail.com	032829730	0995351473	17	

Gráfico 4.18 Diseño de Interfaz Administrar Usuarios
Elaborado por: Andrea Sotelo

1. Menú Principal

Muestra el menú con las opciones y administraciones de acceso con los que cuenta el módulo.

2. Sección de Ingreso Datos

En esta sección el usuario ingresara los datos requeridos para administrar usuarios.

3. Botones de Acciones

Después de haber ingresado los datos podemos realizar varias acciones como Nuevo u Original, Registro, Limpiar, Actualizar y Eliminar.

4. Sección de Información

En esta sección se muestra la información de los Usuarios existentes en donde se puede seleccionar para actualizar o eliminar la información con los Botones de Acciones.

2.2 Administrar Aulas

Nombre	Tipo	cupu	piso	ubicacion	habilitado	codigofacu	id	Seleccionar
AulaF	Aula	30	primero	PRUEBA	<input type="checkbox"/>	1	1	
Aula2	Aula	30	segundo		<input type="checkbox"/>	1	2	
Aula3	Aula	30	tercero		<input type="checkbox"/>	1	3	
Aula4	Aula	30	primero		<input type="checkbox"/>	1	4	
Aula5	Aula	30	segundo		<input type="checkbox"/>	1	5	
Aula6	Aula	30	tercero		<input type="checkbox"/>	1	6	
Aula7	Aula	30	primero		<input type="checkbox"/>	1	7	
Aula8	Aula	30	segundo		<input type="checkbox"/>	1	8	
Aula9	Aula	30	tercero		<input type="checkbox"/>	1	9	
Aula10	Aula	30	primero		<input type="checkbox"/>	1	10	
Aula11	Aula	30	segundo		<input type="checkbox"/>	1	11	
Aula12	Aula	30	tercero		<input type="checkbox"/>	1	12	
Aula13	Aula	30	primero		<input type="checkbox"/>	1	13	

Gráfico 4.19 Diseño de Interfaz Administrar Aulas
Elaborado por: Andrea Sotelo

1. Menú Principal

Muestra el menú con las opciones y administraciones de acceso con los que cuenta el módulo.

2. Sección de Ingreso Datos

En esta sección el usuario ingresara los datos requeridos para administrar aulas.

3. Botones de Acciones

Después de haber ingresado los datos podemos realizar varias acciones como Nuevo, Guardar, Actualizar y Eliminar.

4. Sección de Información

En esta sección se muestra la información de las Aulas registradas en la base de datos en donde se puede seleccionar para actualizar o eliminar la información con los Botones de Acciones.

2.3 Administración de Malla

1

2

3

4

Gráfico 4.20 Diseño de Interfaz Administrar Malla
Elaborado por: Andrea Sotelo

1. Menú Principal

Muestra el menú con las opciones y administraciones de acceso con los que cuenta el módulo.

2. Sección de Ingreso Datos

En esta sección el usuario ingresara los datos requeridos para administrar malla.

3. Botones de Acciones

Después de haber ingresado los datos podemos realizar varias acciones como Nuevo, Guardar, Actualizar y Eliminar.

4. Sección de Información

En esta sección se muestra la información de la malla registrada en la base de datos en donde se puede seleccionar para actualizar o eliminar la información con los Botones de Acciones.

2.4 Disponibilidad Profesor

The screenshot shows the 'Disponibilidad Docente' form with the following fields:

- Docente: [Dropdown]
- Periodo: [Text]
- Hora de Inicio: [07:00:00]
- Hora Fin: [22:00:00]
- Jornada: [Mañana]
- Dia: [Lunes]

Buttons: Nuevo, Guardar, Actualizar

Docente	periodo	Horainicio	Horafin	Jornada	Dia	Seleccionar	Eliminar
	1	09:00:00	10:00:00	Mañana	Lunes	[Icon]	[Icon]
	1	10:00:00	11:00:00	Mañana	Lunes	[Icon]	[Icon]
Darwin Aldás	4	07:00:00	22:00:00	Mañana	Lunes	[Icon]	[Icon]

Gráfico 4.21 Diseño de Interfaz Disponibilidad Docentes
Elaborado por: Andrea Sotelo

1. Menú Principal

Muestra el menú con las opciones y administraciones de acceso con los que cuenta el módulo.

2. Sección de Ingreso Datos

En esta sección el usuario ingresara los datos requeridos para Disponibilidad de Docentes.

3. Botones de Acciones

Después de haber ingresado los datos podemos realizar varias acciones como Nuevo, Guardar, Actualizar y Eliminar.

4. Sección de Información

En esta sección se muestra la información de la disponibilidad de docentes registrada en la base de datos en donde se puede seleccionar para actualizar o eliminar la información con los Botones de Acciones.

2.5 Distributivos

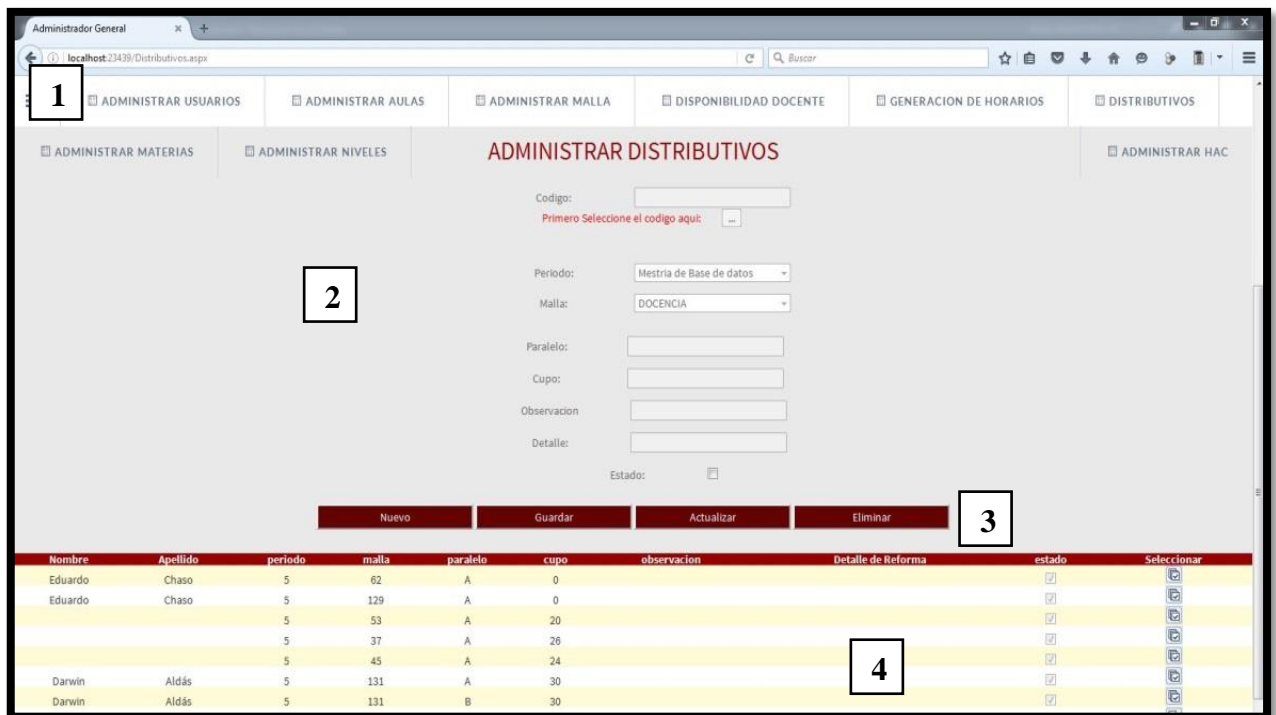


Gráfico 4.22 Diseño de Interfaz Distributivos
Elaborado por: Andrea Sotelo

1. Menú Principal

Muestra el menú con las opciones y administraciones de acceso con los que cuenta el módulo.

2. Sección de Ingreso Datos

En esta sección el usuario ingresara los datos requeridos para distributivos.

3. Botones de Acciones

Después de haber ingresado los datos podemos realizar varias acciones como Nuevo, Guardar, Actualizar y Eliminar.

4. Sección de Información

En esta sección se muestra la información de los distributivos registrada en la base de datos en donde se puede seleccionar para actualizar o eliminar la información con los Botones de Acciones.

2.6 Generación de Horarios



Gráfico 4.23 Diseño de Interfaz Generación de Horarios
Elaborado por: Andrea Sotelo

1. Menú Principal

Muestra el menú con las opciones y administraciones de acceso con los que cuenta el módulo.

2. Sección de Ingreso Datos

En esta sección en la cual se genera los horarios.

4.12 Implementación

En el proceso de implementación del módulo se puede observar lo siguiente:

Conexión a la base de Datos

```
<configuration>
  <connectionStrings>
    <add name="RegistroDocenteConnectionString" connectionString="Data Source=_____12;Initial Catalog=Registro_____
      Integrated Security=True providerName="System.Data.SqlClient" />
    <add name="RegistroDocenteConnectionString2" connectionString="Data Source=_____12;Initial Catalog=Registro_____
      Password=_____lo providerName="System.Data.SqlClient" />
    <add name="RegistroDocenteConnectionString3" connectionString="Data Source=_____12;Initial Catalog=Registro_____
      Password=_____lo providerName="System.Data.SqlClient" />
    <add name="ResBDDConnectionString" connectionString="Data Source=_____12;Initial Catalog=registro_____
      Password=_____lo providerName="System.Data.SqlClient" />
    <add name="RegistroDocenteConnectionString4" connectionString="Data Source=_____12;Initial Catalog=Registro_____
      Password=_____lo providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5" />

    <pages><controls><add tagPrefix="ajaxToolkit" assembly="AjaxControlToolkit" namespace="AjaxControlToolkit" /></controls></pages></system.web>

  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
  </appSettings>
</configuration>
```

Gráfico 4.24 Conexión Base de Datos
Elaborado por: Andrea Sotelo

Algoritmo Genético

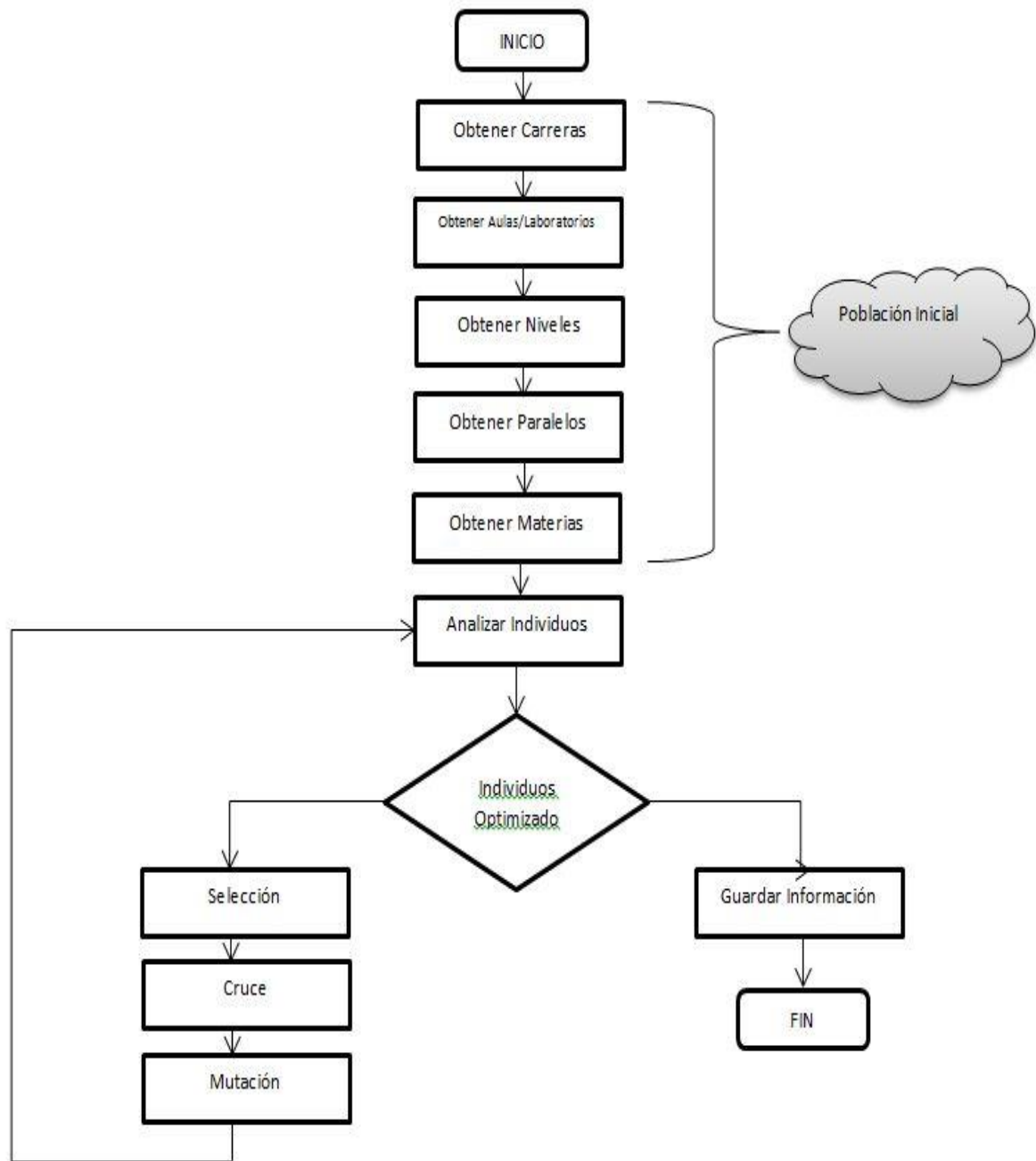


Gráfico 4.25 Diagrama de Flujo Algoritmo Genético
Elaborado por: Andrea Sotelo

Método Información Aulas

```
void Cargar_Datos_Aulas()
{
    try
    {
        //creando la conexion

        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
System.Configuration.ConfigurationManager.ConnectionStrings["RegistroDocente1ConnectionString2"].Conn
ectionString;
        //abriendo conexion

        conn.Open();

        SqlCommand comando = new SqlCommand("Select Nombre, Tipo, cupo, piso, ubicacion, habilitado,
codigofacu from aulas", conn);

        //ejecuta una instruccion de sql devolviendo el numero de las filas afectadas

        comando.ExecuteNonQuery();

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(comando);

        //Llenando el dataAdapter

        da.Fill(ds, "aulas");

        //utilizado para representar una fila de la tabla q necesitas en este caso usuario

        DataRow DR;

        DR = ds.Tables["aulas"].Rows[0];

        //grv_imagenes.SelectedRow.Cells[4].Text = img_mostrar.Visible.ToString();

        GridView_aulas.DataBind();

        comando.ExecuteNonQuery();
    }
    catch
    {
        lbl_error.Text = "Error";
    }
}
```

Método Información Malla

```
void Cargar_Datos_Malla()
{
    try
    {
        //creando la conexion
```

```

        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
System.Configuration.ConfigurationManager.ConnectionStrings["RegistroDocente1ConnectionString2"].Conn
ectionString;
        //abriendo conexion

        conn.Open();

        SqlCommand comando = new SqlCommand("Select m_tipo, carrera, m_materia, nivel, periodo from
malla", conn);

        //ejecuta una instruccion de sql devolviendo el numero de las filas afectadas

        comando.ExecuteNonQuery();

        DataSet ds = new DataSet();

        SqlDataAdapter da = new SqlDataAdapter(comando);

        //Llenando el dataAdapter

        da.Fill(ds, "malla");

        //utilizado para representar una fila de la tabla q necesitas en este caso usuario

        DataRow DR;

        DR = ds.Tables["malla"].Rows[0];

        //grv_imagenes.SelectedRow.Cells[4].Text = img_mostrar.Visible.ToString();

        GridView_malla.DataBind();

        comando.ExecuteNonQuery();
    }
    catch
    {
        lbl_error.Text = "Error";
    }
}

```

Método Información Disponibilidad

```

void Cargar_Datos_Disponibilidad()
{
    try
    {
        //creando la conexion

        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
System.Configuration.ConfigurationManager.ConnectionStrings["RegistroDocente1ConnectionString2"].Conn
ectionString;
        //abriendo conexion

```

```

conn.Open();

SqlCommand comando = new SqlCommand("Select adusu_codigo, adusu_usuario, periodo,
HoraInicio, HoraFin, Jornada, Dia from mdisponibilidad", conn);

//ejecuta una instruccion de sql devolviendo el numero de las filas afectadas
comando.ExecuteNonQuery();

DataSet ds = new DataSet();

SqlDataAdapter da = new SqlDataAdapter(comando);

//Llenando el dataAdapter
da.Fill(ds, "mdisponibilidad");

//utilizado para representar una fila de la tabla q necesitas en este caso usuario
DataRow DR;

DR = ds.Tables["mdisponibilidad"].Rows[0];

//grv_imagenes.SelectedRow.Cells[4].Text = img_mostrar.Visible.ToString();

GridView_distributivo_principal.DataBind();

comando.ExecuteNonQuery();
}
catch
{
    lbl_error1.Text = "Error";
}
}

```

Generación Algoritmo

```

DateTime oldDate = DateTime.Now;
lbl_resultado_time.Text = "Ejecutando Iteraciones";
if (Convert.ToInt32(txt_iteraciones.Text) >= 0)
{
    for (int a = 0; a <= Convert.ToInt32(txt_iteraciones.Text); a++)
    {

        NewDatosTableAdapters.Vis_CarrerasTableAdapter dtcarrera = new
NewDatosTableAdapters.Vis_CarrerasTableAdapter();
        NewDatos.Vis_CarrerasDataTable tacarrera = new NewDatos.Vis_CarrerasDataTable();
        dtcarrera.Fill(tacarrera);

        NewDatosTableAdapters.aulasTableAdapter dtaulas = new
NewDatosTableAdapters.aulasTableAdapter();
        NewDatos.aulasDataTable taaulas = new NewDatos.aulasDataTable();
        dtaulas.Fill(taaulas);
    }
}

```

```

int aula_inicio = Convert.ToInt32(taaulas.Rows[0]["id_aula"].ToString());
int aula_fin = Convert.ToInt32(taaulas.Rows[taaulas.Rows.Count - 1]["id_aula"].ToString());

DataTable calendario = new DataTable();
calendario.Columns.Add("dia");
calendario.Columns.Add("hora");
calendario.Columns.Add("materia");
calendario.Columns.Add("aula");
calendario.Columns.Add("nivel");
calendario.Columns.Add("carrera");
calendario.Columns.Add("paralelo");

for (int s = 0; s <= tacarrera.Rows.Count - 1; s++)
{
    NewDatosTableAdapters.nivelesTableAdapter dtniveles = new
NewDatosTableAdapters.nivelesTableAdapter();
    NewDatos.nivelesDataTable taniveles = new NewDatos.nivelesDataTable();
    dtniveles.FillBy(taniveles, Convert.ToInt32(tacarrera.Rows[s]["carrera"].ToString()));

    for (int r = 0; r <= taniveles.Rows.Count - 1; r++)
    {
        NewDatosTableAdapters.Vis_ParalelosTableAdapter dtparalelos = new
NewDatosTableAdapters.Vis_ParalelosTableAdapter();
        NewDatos.Vis_ParalelosDataTable taparalelos = new NewDatos.Vis_ParalelosDataTable();
        dtparalelos.FillBy(taparalelos, Convert.ToInt32(taniveles.Rows[r]["id_nivel"].ToString()));

        for (int q = 0; q <= taparalelos.Rows.Count - 1; q++)
        {
            NewDatosTableAdapters.horariosTableAdapter dthorarios = new
NewDatosTableAdapters.horariosTableAdapter();
            NewDatos.horariosDataTable tahorarios = new NewDatos.horariosDataTable();
            dthorarios.FillByJornada(tahorarios,
Convert.ToInt32(taniveles.Rows[r]["jornada"].ToString()));
            int horario_inicio = Convert.ToInt32(tahorarios.Rows[0]["id_horario"].ToString());
            int horario_fin = Convert.ToInt32(tahorarios.Rows[tahorarios.Rows.Count -
1]["id_horario"].ToString());

            NewDatosTableAdapters.Vis_materiasTableAdapter dtmaterias = new
NewDatosTableAdapters.Vis_materiasTableAdapter();
            NewDatos.Vis_materiasDataTable tamaterias = new NewDatos.Vis_materiasDataTable();
            dtmaterias.FillByNivel(tamaterias,
Convert.ToInt32(taniveles.Rows[r]["id_nivel"].ToString()));
            int materia_inicio = Convert.ToInt32(tamaterias.Rows[0]["id_materia"].ToString());
            int materia_fin = Convert.ToInt32(tamaterias.Rows[tamaterias.Rows.Count -
1]["id_materia"].ToString());

            for (int h = 0; h <= tamaterias.Rows.Count - 1; h++)
            {
                int horas = Convert.ToInt32(tamaterias.Rows[h]["horasSemanal"].ToString());
                for (int g = 0; g <= horas - 1; g++)
                {
                    bool seguir = false;
                    while (seguir == false)
                    {
                        Random rnd = new Random();
                        int dia = rnd.Next(1, 6);
                        int horario = rnd.Next(horario_inicio, horario_fin);
                        int materia = Convert.ToInt32(tamaterias.Rows[h]["id_materia"].ToString());
                        int aula = rnd.Next(aula_inicio, aula_fin);
                        int nivel = Convert.ToInt32(taniveles.Rows[r]["id_nivel"].ToString());
                        int carrera = Convert.ToInt32(tacarrera.Rows[s]["carrera"].ToString());
                    }
                }
            }
        }
    }
}

```



```

for (int j = 0; j <= dtlNivel.Items.Count - 1; j++)
{
    DataList dtlParalelo = (DataList)dtlNivel.Items[j].FindControl("dlParalelo");
    Label nvl = (Label)dtlNivel.Items[j].FindControl("lblCodNivel");

    NewDatosTableAdapters.Vis_ParalelosTableAdapter dtparalelos = new
NewDatosTableAdapters.Vis_ParalelosTableAdapter();
    NewDatos.Vis_ParalelosDataTable taparalelos = new NewDatos.Vis_ParalelosDataTable();
    dtparalelos.FillBy(taparalelos, Convert.ToInt32(nvl.Text));

    dtlParalelo.DataSource = taparalelos;
    dtlParalelo.DataBind();

    for (int k = 0; k <= dtlParalelo.Items.Count - 1; k++)
    {
        GridView gv = (GridView)dtlParalelo.Items[k].FindControl("gvCalendario");
        Label pr = (Label)dtlParalelo.Items[k].FindControl("lblCodParalelo");
        DataTable dt = new DataTable();
        dt.Columns.Add("Hora/Dia");
        dt.Columns.Add("Lunes");
        dt.Columns.Add("Martes");
        dt.Columns.Add("Miercoles");
        dt.Columns.Add("Jueves");
        dt.Columns.Add("Viernes");
        DataView dv = calendario.DefaultView;
        NewDatosTableAdapters.horariosTableAdapter dthorarios = new
NewDatosTableAdapters.horariosTableAdapter();
        NewDatos.horariosDataTable tahorarios = new NewDatos.horariosDataTable();
        dthorarios.FillByJornada(tahorarios,
Convert.ToInt32(taniveles.Rows[j]["jornada"].ToString()));
        for (int l = 0; l <= tahorarios.Rows.Count - 1; l++)
        {
            DataRow dr = dt.NewRow();
            for (int m = 0; m <= taulas.Rows.Count - 1; m++)
            {
                for (int n = 1; n <= 5; n++)
                {
                    dv.RowFilter = "carrera=" + cr.Text + " and nivel=" + nvl.Text + " and
paralelo=" + pr.Text + " and dia=" + n.ToString() + " and hora=" +
tahorarios.Rows[l]["id_horario"].ToString() + " and aula=" + taulas.Rows[m]["id_aula"].ToString() + "";

                    dr["Hora/Dia"] = tahorarios.Rows[l]["h_inicio"].ToString() + "/" +
tahorarios.Rows[l]["h_fin"].ToString();
                    if (dv.Count > 0)
                    {
                        NewDatosTableAdapters.Vis_materiasTableAdapter tamat = new
NewDatosTableAdapters.Vis_materiasTableAdapter();
                        dr[n] = tamat.ObtenerMateria(Convert.ToInt32(dv[0]["materia"].ToString()));
                    }
                }
            }
            dt.Rows.Add(dr);
        }
        //btnGenerar.Text = dt.Rows.Count.ToString();
        gv.DataSource = dt;
        gv.DataBind();
    }
}

```

```
    }  
  }  
}  
  
DateTime newDate = DateTime.Now;  
TimeSpan ts = newDate - oldDate;  
lbl_resultado_time.Text = Convert.ToString(ts);
```


4.13 Pruebas

Las pruebas son un aspecto esencial en el desarrollo de software y mucho más en la metodología extreme programming en donde hacemos rectificaciones constantes para poder mejorar el desempeño del software.

Al concluir con cada especificación de desarrollo se reúne el equipo de desarrollo para realizar las pruebas necesarias con el fin de corregir pequeños detalles y concluir la metáfora y continuar con la siguiente refactorización. Las pruebas utilizadas constantemente son pruebas de caja blanca, pruebas de caja negra y pruebas de aceptación.

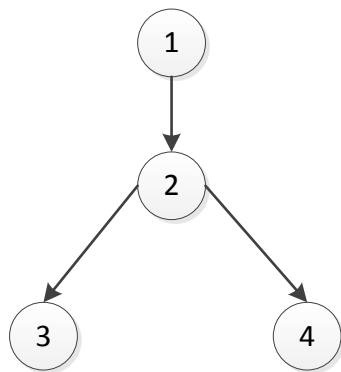
4.13.1 Pruebas de Caja Blanca

Las pruebas de Caja Blanca son utilizadas para verificar que las funciones internas puedan comprobar el control del diseño procedimental y lo que se verifico en el módulo fue:

- Se comprobó el buen funcionamiento de cada elemento existente en la interfaz así como en el menú establecido para la administración de los diferentes opciones del módulo, permitiendo que sea fácil el manejo al usuario.
- Evaluar el control de errores, validaciones, así como la respuesta del módulo en cuanto a selecciones, inserciones, actualizaciones y eliminaciones de cada administración; además de la generación de horarios.

El uso del módulo va estar a cargo de usuarios que laboren en la institución, sin embargo va hacer intuitivo y que permita un manejo sencillo a cada usuario que cuente con el más básico conocimiento tecnológico.

Ingreso al Módulo



- 1) Ingreso Usuario
- 2) Ingreso Clave
- 3) Proceso Exitoso
- 4) Proceso Fallido

Opciones:

Proceso Exitoso: 1,2,3

Proceso Fallido: 1,2,4

Gráfico 4.26 Pruebas de Ingreso al Módulo

Elaborado por: Andrea Sotelo


4.14.2 Pruebas de Caja Negra

Las Pruebas de Caja Negra son utilizadas para las pruebas funcionales operativas en donde las entradas de información se acepten de manera adecuada y que se realiza una salida correcta. Lo que se verificó en el módulo fue:

- Se comprobó el ingreso de la información necesaria de forma correcta.
- Se verificó las entradas y salidas de la información de cada administración.
- Se corroboró el control de Distributivos tanto horizontal como vertical.
- Se evaluó la generación de horarios de acuerdo a los requerimientos establecidos previamente.

Controles de Ingreso

En el Control de Ingreso verifica que sea administrador o que este asignado su rol y su clave sea la correcta.



The image shows a web interface for a system administrator login. The main heading is "Administrador General" in a large, dark red, serif font. Below this is a dark gray rectangular box containing the login form. The form has the following elements:

- The text "Administrador General" in a smaller, dark red font.
- A text input field containing the username "admin".
- A text input field labeled "Contraseña" (Password).
- A button labeled "Acceder" (Access).
- An error message at the bottom: "Error! Su contraseña no es correcta o no es de Administrador".

The interface is set against a light gray background. At the top, there is a navigation bar with a hamburger menu icon on the left and the text "ADMINISTRADOR" on the right.

Gráfico 4.27 Diseño de Interfaz Ingreso Sistema

Elaborado por: Andrea Sotelo

Administrar Usuario

Administrador General

localhost:23439/Registro_usuarios.aspx

ADMINISTRAR USUARIOS ADMINISTRAR AULAS ADMINISTRAR MALLA DISPONIBILIDAD DOCENTE GENERACION DE HORARIOS DISTRIBUTIVOS

ADMINISTRAR MATERIAS ADMINISTRAR NIVELES ADMINISTRAR USUARIOS ADMINISTRAR HAC

Codigo:

Nombre:

Usuario:

Rol:

Contraseña:

Repita la Contraseña:

Email:

Teléfono Convencional:

Teléfono Celular:

Falta rellenar o editar los siguientes campos:
Debe ingresar el código
Debe ingresar el nombre
Debe ingresar el usuario
Debe ingresar su correo
Debe ingresar el número de teléfono convencional
Debe ingresar su número móvil

Ver Usuarios Original Regístrate Limpiar Actualizar Eliminar

Gráfico 4.28 Diseño de Interfaz Administrar Usuario
Elaborado por: Andrea Sotelo

Administrar Aulas

Administrador General

localhost:23439/Administrar_aulas.aspx

ADMINISTRAR USUARIOS ADMINISTRAR AULAS ADMINISTRAR MALLA DISPONIBILIDAD DOCENTE GENERACION DE HORARIOS DISTRIBUTIVOS

ADMINISTRAR MATERIAS ADMINISTRAR NIVELES ADMINISTRAR HAC

ADMINISTRAR AULAS

Nombre del Aula:

Tipo:

Cupo:

Piso:

Ubicacion:

Habilitado:

Facultad:

solo se puede aceptar letras
Solo puede ingresar numeros enteros
Solo puede ingresar letras
Solo puede ingresar letras

Original Guardar Actualizar Eliminar

Nombre	Tipo	cupo	piso	ubicacion	habilitado	codigofacu	id	Seleccionar
AulaF	Aula	30	primero	PRUEBA	<input type="checkbox"/>	1	1	<input type="button" value="Seleccionar"/>
Aula2	Aula	30	segundo		<input type="checkbox"/>	1	2	<input type="button" value="Seleccionar"/>
Aula3	Aula	30	tercero		<input type="checkbox"/>	1	3	<input type="button" value="Seleccionar"/>
Aula4	Aula	30	primero		<input type="checkbox"/>	1	4	<input type="button" value="Seleccionar"/>
Aula5	Aula	30	segundo		<input type="checkbox"/>	1	5	<input type="button" value="Seleccionar"/>
Aula6	Aula	30	tercero		<input type="checkbox"/>	1	6	<input type="button" value="Seleccionar"/>

Gráfico 4.29 Diseño de Interfaz Administrar Aulas
Elaborado por: Andrea Sotelo

Revisa que todos los campos estén llenos con el tipo de datos correctos.

4.14 Implantación

Nº	Proceso	Descripción	Implicados
1	Presentación del Proyecto	Explicación de la etapas del proyecto	Administrador de Redes Equipo de Desarrollo (Andrea Sotelo)
2	Establecer la modificaciones y corrección de errores	Creación de la Lista de cambios y correcciones necesarias.	Administrador de Redes Equipo de Desarrollo (Andrea Sotelo)
3	Presentación del proyecto con los cambios	Verificación de los cambios realizados.	Administrador de Redes Equipo de Desarrollo (Andrea Sotelo)
4	Pruebas necesarias con cada sección del Modulo	Comprobación del funcionamiento del módulo con datos.	Administrador de Redes Equipo de Desarrollo (Andrea Sotelo)
5	Verificación de errores	En el ingreso de datos se identifica los detalles que deben ser cambiados.	Administrador de Redes Equipo de Desarrollo (Andrea Sotelo)
6	Pruebas de Caja Blanca y Negra	Demostración del cumplimiento en las funciones establecidas de cada sección.	Administrador de Redes Equipo de Desarrollo (Andrea Sotelo)
7	Pruebas de Aceptación	Pruebas con datos reales para el Control de Distributivos y Generación de Horarios	Administrador de Redes Equipo de Desarrollo (Andrea Sotelo)
8	Entrega del sistema	Entrega del Sistema al Administrador de Redes por Andrea Sotelo	Administrador de Redes Equipo de Desarrollo (Andrea Sotelo)

4.14.1 Pruebas de Aceptación

La ejecución de esta prueba se la hace con el fin de poder comprobar los requerimientos previamente definidos por el área de Administración de Redes, en donde se hizo presentaciones constantes de los avances del módulo hasta llegar al producto final con las modificaciones respectivas quedando satisfecho de los resultados obtenidos con el módulo.

Con el Producto Final se ejecutó las pruebas respectivas para la verificación del correcto funcionamiento con el Administrador de Redes, de modo que cada Sección del Módulo cumpla con sus funciones respectivas.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Por medio de la elaboración de encuestas se pudo obtener la información necesaria para elegir las herramientas necesarias para el desarrollo del Módulo e identificar los requerimientos que fueron establecidos por la Administración de Redes.
- Los complementos utilizados en el desarrollo como JQuery, Ajax, JavaScript, ItextSharp permitieron que el diseño de interfaz sea sencillo y de fácil manejo para el usuario
- Con la Integración de la Disponibilidad de los docentes al Modulo permite el acceso a esa información que es necesaria para la Generación de Horarios
- Mediante la Generación de Horarios se optimizo el tiempo y permitió que no exista anomalías en los datos utilizados en los Horarios demostrado anteriormente en los anexos expuestos.
- Con la Implantación del Módulo de Control de Distributivos y Generación de Horarios los procesos que intervienen en ellos son mucho más ágiles, la información es mucho más segura además solo los usuarios con permisos pueden acceder a esta información; creando confiabilidad en los datos que se obtiene en un reporte.

- La utilización de Algoritmos Genéticos permitieron que el desarrollo de software en cuanto a Generación de horarios llegue a cumplir los requisitos establecidos para el Modulo.
- Con el funcionamiento de Algoritmos Genéticos se realiza un determinado número de iteraciones necesarias que permite que el resultado obtenido en cuanto a horarios sea mejor en cada generación.
- Los Algoritmos Genéticos permiten tomar de cada generación lo resultados más óptimos convirtiéndolos en una nueva población para otra generación si así se lo desea obteniendo de esta manera horarios más adecuados

5.2 RECOMENDACIONES

- Se recomienda que el Administrador de Redes realice backups de la base de datos siendo lo más adecuado cada trimestre debido a que no se realizan muchas modificaciones en la Información de Distributivos y Horarios.
- La capacitación para el manejo del módulo es esencial, para saber la importancia de cada sección e incluso la manera de ingreso de los datos además del funcionamiento adecuado del módulo y los procesos que realiza cada sección.
- Se debe configurar los navegadores en los cuales se pueda ejecutar el Modulo ya que utiliza código JQuery, Ajax, JavaScript y estos son necesarios para control de elementos y diseño de interfaz. De esta manera permitiendo el buen funcionamiento del módulo.
- Se recomienda profundizar un poco más en cuanto a algoritmos genéticos, ya que son adaptables a varios posibles soluciones en diferentes problemáticas; haciendo que sea mucho más eficientes sus resultados y con mejoras constantes.
- Es necesario que en el proceso de codificación de Algoritmos Genéticos se pueda definir bien el tamaño de la población así como la función de aptitud para que los resultados obtenidos sean los más óptimos.
- El modulo se lo puede ir desarrollando en proyectos futuros, además de que pueden ser aplicables para cada facultad de la universidad.

BIBLIOGRAFÍA

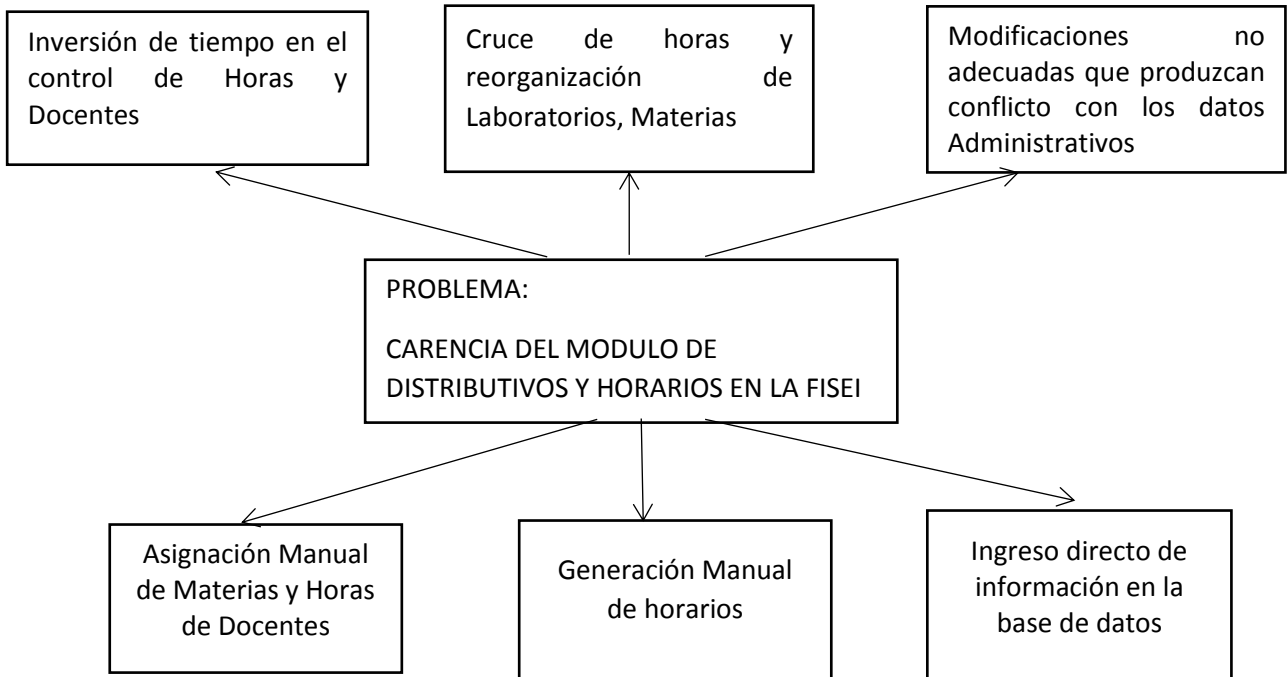
- [1] R. G.-M. F. V. M. Horacio Kuna, «PROCEDIMIENTOS DE LA EXPLOTACIÓN DE INFORMACIÓN PARA LA IDENTIFICACIÓN DE DATOS FALTANTES, CON RUIDO E INCONSISTENTES,» Universidad Nacional de Misiones, Universidad de Buenos Aires, Universidad de Málaga.
- [2] J. A. E. García, AUDITORÍA EN INFORMÁTICA, Mexico: McGraw-Hill, 2001.
- [3] P. L. y. M. C. P. José H. Canós, «Metodologías Ágiles en el Desarrollo de Software,» Universidad Politécnica de Valencia, Valencia, 2006.
- [4] M. C. M. Stalin, «SISTEMA AUTOMATIZADO PARA EL REGISTRO Y CONTROL DOCENTE DE LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL APLICANDO NUEVAS TECNOLOGÍAS,» UNIVERSIDAD TÉCNICA DE AMBATO, AMBATO, 2012.
- [5] L. A. L. V. Sonia Edith Barreto Barros, «ALGORITMO PARA GESTIÓN DE HORARIOS DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD DE CUENCA,» UNIVERSIDAD DE CUENCA, CUENCA, 2013.
- [6] J. R. C. C. EDISON GUSTAVO CAIZA ANDRANGO, «DESARROLLO DE UN AGENTE DE SOFTWARE PARA LA GENERACIÓN AUTOMÁTICA DE HORARIOS, UTILIZANDO LA METODOLOGÍA AUML, EN UN SISTEMA ESCOLÁSTICO PARAMETRIZABLE, VÍA WEB, APLICANDO SOFTWARE LIBRE, PARA SU IMPLANTACIÓN EN LA UNIDAD EDUCATIVA HERMANO MIGUEL – LATACU,» ESCUELA POLITÉCNICA DEL EJÉRCITO EXTENSIÓN LATACUNGA, LATACUNGA, 2012.
- [7] I. G. P. Arias, «SISTEMA INTELIGENTE DE SOPORTE EN LA GENERACIÓN DE HORARIOS ACADÉMICOS PARA LA CARRERA DE INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA,» UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO, QUITO, 2010.
- [8] S. L. Mora, «Programación de aplicaciones web: historia, principios básicos y clientes web,» Alicante , 2002.
- [9] M. Á. S. U. D. R. G. Salvador Sánchez Alonso, Ingeniería del Software: Un enfoque desde la guía SWEBOK, México: Grupo Editorial Gaceta, 2012.
- [10] R. S. Pressman, INGENIERÍA DEL SOFTWARE UN ENFOQUE PRÁCTICO Quinta edición, Madrid: McGRAW-HILL, 2002.
- [11] I. SOMMERVILLE, Ingeniería del software Séptima edición, Madrid: PEARSON EDUCACIÓN. S.A., 2005.
- [12] R. S. Pressman, INGENIERÍA DEL SOFTWARE UN ENFOQUE PRÁCTICO Séptima edición,

-] Madrid: McGRAW-HILL, 2010.
- [13 S. C. y. T. G. César Camisón, GESTIÓN DE LA CALIDAD: CONCEPTOS, ENFOQUES, MODELOS Y SISTEMAS, Madrid: PEARSON EDUCACIÓN. S.A., 2006.
- [14 M. K. y. S. S. Mary Beth Chrissis, «Guía para la integración de procesos y la mejora de productos DE CUENCA,» Universidad Politécnica de Madrid, Madrid, 2009.
- [15 S. J. R. y. P. Norvig, INTELIGENCIA ARTIFICIAL UN ENFOQUE MODERNO, Madrid: PEARSON EDUCACIÓN, S.A., 2004.
- [16 J. T. P. M. y. R. M. MORALES, INTELIGENCIA ARTIFICIAL: Métodos, técnicas y aplicaciones, Madrid: McGRAW-HILL, 2008.
- [17 A. T. F. Q. A. S. E. M. A. C. E. S. Y. T. M. D. T. F. O. J. H. C. Z. N. V. y. O. P. Julio Ponce, Inteligencia Artificial, Latinoamerica: Iniciativa Latinoamericana de Libros de Texto Abiertos, 2014.
- [18 D. P. P. Cruz, INTELIGENCIA ARTIFICIAL CON APLICACIONES A LA INGENIERÍA, México D.F.: Alfaomega Grupo Editor, S.A., 2010.
- [19 J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
- [20 A. Marczyk, «The TalkOrigins Archive Exploring the Creation/Evolution Controversy,» 23 04 2004. [En línea]. Available: <http://www.talkorigins.org/faqs/genalg/genalg.html>.
- [21 N. A. y. J. A. C. Jonás A. Montilva, «Desarrollo de Software Basado en Componentes,» Universidad de Los Andes, Mérida, 2003.
- [22 U. T. d. Ambato, «Universidad Técnica de Ambato,» [En línea]. Available: <http://www.uta.edu.ec/v2.0/pdf/uta/distributivotrabajo.pdf>. [Último acceso: 22 12 2014].
- [23 Microsoft, «Microsoft Developer Network,» [En línea]. Available: [https://msdn.microsoft.com/es-es/library/4w3ex9c2\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx). [Último acceso: 2016 01 06].
- [24 Microsoft, «Microsoft Developer Network,» [En línea]. Available: [http://msdn.microsoft.com/es-es/library/4w3ex9c2\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx). [Último acceso: 2014 10 09].
- [25 Microsoft, «Microsoft Developer Network,» [En línea]. Available: <https://msdn.microsoft.com/es-es/library/bb545450.aspx>. [Último acceso: 2016 01 06].
- [26 P. M. Bowman, «QA En dispositivos moviles,» [En línea]. Available: <https://qaendispositivosmoviles.wordpress.com/2013/05/04/que-es-y-para-que-sirve-jquery-y-html5/>. [Último acceso: 2016 01 07].
- [27 E. González, «aprenderaprogramar.com,» [En línea]. Available:

-] http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=590:i-que-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192.
[Último acceso: 2016 01 07].
- [28 R. Torres, «desarrolladores.me,» [En línea]. Available:
] <http://desarrolladores.me/2014/01/c-creando-archivos-pdf-con-itextsharp/>.
- [29 P. L. y. M. C. P. José H. Canós, «Métodologías Ágiles en el Desarrollo de Software,»
] Universidad Politécnica de Valencia, Valencia, 2004.
- [30 Microsoft, «Microsoft Developer Network,» [En línea]. Available:
] [http://msdn.microsoft.com/es-es/library/4w3ex9c2\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx). [Último acceso: 07 10 2014].
- [31 M. K. y. S. S. Mary Beth Chrissis, «Guía para la integración de procesos y la mejora de
] productos,» Universidad Politécnica de Madrid, Madrid, 2009.

ANEXOS

Anexo 1



Anexo 2



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS,
ELECTRÓNICA e INDUSTRIAL
Cda. Universitaria (Predios Huachi)
Casilla 334 / Telefax: 03-2851894
email: fts@uta.edu.ec / fisei@uta.edu.ec
AMBATO - ECUADOR



Ambato, 08 de Septiembre del 2015
FISEI-AR-090-2015

Ingeniero
Vicente Morales
DECANO
FACULTAD DE INGENIERIA EN SISTEMAS ELECTRONICA E INDUSTRIAL
UNIVERSIDAD TÉCNICA DE AMBATO
Presente

De mis consideraciones

En atención a la sumilla del oficio FISEI-D-1556 donde se adjunta documentación con fines de elaboración de Horarios del Personal Docente de la FISE, tengo a bien solicitar se realice una revisión a los distributivos para que estos concuerden en las carreras de Electrónica e Industrial.

Por la atención que se digne dar a la presente agradezco y suscribo

Atentamente,

Ing. Eduardo Chaso
ADMINISTRADOR DE REDES Y SISTEMAS
Facultad de Ingeniería en Sistemas,
Electrónica e Industrial



RECIBIDO 08 SEP 2015
DCA.00

Anexo 3



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS,
ELECTRÓNICA e INDUSTRIAL
Cda. Universitaria (Predios Huachi)
Casilla 334 / Telefax: 03-2851894
email: fis@uta.edu.ec / fisei@uta.edu.ec
AMBATO - ECUADOR



Ambato, 20 de Octubre del 2015

FISEI-AR-111-2015

Ingeniero Mg.

Vicente Morales

DECANO

FACULTAD DE INGENIERIA EN SISTEMAS ELECTRONICA E INDUSTRIAL

UNIVERSIDAD TÉCNICA DE AMBATO

Presente

De mis consideraciones:

La presente tiene como fin poner a consideración el informe de asistencias de docentes desde el día martes 13 de Octubre hasta el día viernes 16 de octubre del presente año debido al cambio del sistema registro control docente que se dio en la Universidad, de igual manera adjunto el registro de asistencia del día 12 de Octubre debido a un corte de luz suscitado a las 7:15am, además los reportes de fallo en el sistema de ciertos docentes los días 05,07 y 08 de Octubre, esto en virtud que la persona encargada se encontraba en periodo de vacaciones.

Atentamente,

Ing. Eduardo Chaso
ADMINISTRADOR DE REDES
Facultad de Ingeniería en Sistemas,
Electrónica e Industrial



Adj:63 hojas



RECIBIDO 20 OCT 2015
Morales
12400

Anexo 4

Generación de Horarios



Anexo 5



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS,
ELECTRÓNICA e INDUSTRIAL
Cda. Universitaria (Predios Huachi)
Casilla 334 / Telefax: 03-2851894
email: fis@uta.edu.ec / fiset@uta.edu.ec
AMBATO - ECUADOR



Ambato, 30 de Marzo del 2016
FISEI-AR-043-A-2016

Ingeniero Mg.
Mario García
DECANO (E)
FACULTAD DE INGENIERIA EN SISTEMAS ELECTRONICA E INDUSTRIAL
UNIVERSIDAD TÉCNICA DE AMBATO
Presente

De mis consideraciones:

La presente tiene como fin informar que el trabajo de investigación titulado: *"Módulo de Control de Distributivos y Generación de Horarios Integrado al Sistema de Control de Docentes de la FISEI en la Universidad Técnica de Ambato"* desarrollado por el Srta. Sotelo Villalva Andrea Elizabeth, ha sido concluido en su etapa de desarrollo tomado en cuenta que se lo ha realizado con datos ficticios ingresados por la Srta., el departamento iniciara la etapa de pruebas de caja negra en los siguiente días y así poder a futuro dar uso a lo desarrollado.

Por la favorable atención que se digne dar a la presente agradezco y suscribo.

Atentamente,

Ing. Eduardo Chaso
ADMINISTRADOR DE REDES
Facultad de Ingeniería en Sistemas,
Electrónica e Industrial

