



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
UNIDAD DE ESTUDIOS A DISTANCIA
MODALIDAD SEMIPRESENCIAL
CARRERA DE SISTEMAS
TESIS DE GRADO

TEMA

“Aplicación web para el control de la gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo”.

AUTOR

WILMER ESTUARDO CEVALLOS MUÑOZ

DIRECTOR

ING. CRISTIAN GABRIEL ZAMBRANO VEGA

QUEVEDO – LOS RIOS – ECUADOR

2015

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **WILMER ESTUARDO CEVALLOS MUÑOZ**, declaro que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

f. _____

WILMER ESTUARDO CEVALLOS MUÑOZ

C. I. 120188907-6

CERTIFICACIÓN DEL DIRECTOR DE TESIS

El suscrito, Ing. Cristian Gabriel Zambrano Vega, Docente de la Unidad de Estudios a Distancia de la Universidad Técnica Estatal de Quevedo, certifico que el Egresado Wilmer Estuardo Cevallos Muñoz, realizó la Tesis de Grado titulada “**APLICACIÓN WEB PARA EL CONTROL DE LA GESTIÓN DE LAS ACTIVIDADES COMPLEMENTARIAS DE LOS DOCENTES DE LA UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO**”, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

ING. CRISTIAN GABRIEL ZAMBRANO VEGA
DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
UNIDAD DE ESTUDIOS A DISTANCIA
MODALIDAD SEMIPRESENCIAL
CARRERA DE SISTEMAS

TESIS DE GRADO

**“APLICACIÓN WEB PARA EL CONTROL DE LA GESTIÓN DE
LAS ACTIVIDADES COMPLEMENTARIAS DE LOS DOCENTES
DE LA UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO”**

Presentada al Honorable Comité Técnico Administrativo de la Unidad de Estudios a Distancia como requisito previo a la obtención del título de INGENIERIA EN SISTEMAS.

MIEMBROS DEL TRIBUNAL

LCDO. FREDDY CHAMORRO PALACIOS

LCDO. JUAN CARLOS ORTEGA ACOSTA

ING. RICARDO AGUIRRE PEREZ Msc.

Quevedo- Los Ríos - Ecuador

2015

DEDICATORIA

Con mucho amor y respeto;
A la memoria de mis padres, Rafael e Iluminada,
A mi Sra. Esposa Mercy,
A mi querido hermano Omar por su convicción y
A mis amados hijos Adrián, Sheyla, Wilmer y Jessy.

AGRADECIMIENTO

A Dios eternamente

A mi familia por su apoyo y paciencia

A mis compañeros hermanos

A mis amigos docentes por su confianza y estima

Y a todas las personas que de alguna manera

Influyeron en la realización de este proyecto.

INDICE

Portada	i
Declaración de Autoría y Cesión de Derechos.....	ii
Certificación del Director de Tesis.....	iii
Portada Miembros del Tribunal	iv
Dedicatoria.....	v
Agradecimiento	vi
Índice.....	vii
Índice de Ilustraciones.....	xii
Índice de Gráficos.....	xiii
Índice de Cuadros.....	xiv
Resumen Ejecutivo.....	xvi
Executive Summary.....	xvii
CAPITULO I.....	1
1.1. Introducción	2
1.2. Objetivos.....	4
1.2.1. General	4
1.2.2. Específicos:.....	4
1.3. Hipótesis.....	5
CAPITULO II.....	6
2.1. Fundamentación Teórica	7
2.1.1. Metodologías Ágiles en el Desarrollo de Software	7
2.1.1.1. Metodología Kanban.....	7
2.1.1.2. Metodología Scrum.....	8
2.1.1.3. Metodología XP (Xtreme Programming)	8
2.1.1.4. Metodología RAD (Rapid Application Development)	9

2.1.2.	Aplicaciones web	12
2.1.3.	Diagrama de casos de uso.....	13
2.1.4.	Diagrama de actividades.....	13
2.1.5.	Diagrama de secuencias.....	14
2.1.6.	Diagrama de clases	14
2.1.7.	Lenguaje de programación C#	14
2.1.8.	Programación Orientada a Objetos (POO).....	14
2.1.9.	Arquitectura ASP.NET.....	15
2.1.10.	Modelo vista controlador (MVC).....	16
2.1.11.	Modelo CodeFirst Basado en Entity Framework	17
2.1.12.	Generalidades de Entity Framework	25
2.1.13.	Microsoft Project	28
2.1.14.	Motor Razor	28
2.1.15.	JavaScript	29
2.1.16.	JQuery.....	31
2.1.17.	Ajax	32
2.1.18.	BootStrap 3.0	35
2.1.19.	Html5.....	35
2.1.20.	Visual Studio	37
2.1.21.	Regular Expressions	38
2.1.22.	JSon	39
2.1.23.	DBMS.....	40
2.1.24.	RDBMS	45
2.1.25.	SQL Server	46
2.1.26.	Common Language Runtime (CLR).....	47
2.1.27.	Data Warehouse	47
2.1.28.	Norma IEEE	48
2.1.29.	Actividades Complementarias.....	49
2.1.30.	UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO (UTEQ)	50
CAPITULO III.....		52
3.1.	Ubicación	53
3.2.	Materiales y métodos.....	53
3.2.1.	Materiales y Equipos.....	53
3.3.	Método.....	54

3.3.1.	Inductivo- deductivo:	54
3.3.2.	Descriptivo:	54
3.3.3.	Analítico o Explicativo:	54
3.4.	Tipo de Investigación	54
3.4.1.	De campo y exploratoria.	55
3.4.2.	Identificación del problema.....	55
3.5.	Fuentes.....	56
3.5.1.	Primarias.....	56
3.5.2.	Secundarias y electrónicas	56
3.6.	Población y muestra	58
3.6.1.	Población	58
3.6.2.	Muestra	58
3.7.	Procedimiento Metodológico.....	59
3.7.1.	Fase de determinación de requerimientos	60
3.7.2.	Fase de análisis de la estructura del modelo de datos	61
3.7.3.	Fase de definición del tipo de arquitectura de desarrollo	61
3.7.4.	Fase de diseño.....	61
3.7.5.	Fase de Desarrollo	62
3.7.6.	Fase de Implementación	62
CAPITULO IV.....		64
4.1.	Análisis de Resultados.....	65
4.2.	Discusión	71
4.3.	Análisis de situación actual.....	73
4.3.1.	Identificación de los factores que se involucran en el problema	73
4.3.2.	Factores controlables y no controlables	74
4.3.2.1.	Factores Controlables:.....	74
4.3.2.2.	Factores no controlables:.....	75
4.3.3.	Descripción del problema en relación a los factores.....	76
4.3.4.	Planteamiento relacional o funcional del problema.....	76
4.3.5.	Bibliográfica	77
4.3.6.	Análisis del proceso anterior de registro y control de Actividades Complementarias.....	78
4.3.7.	Análisis del proceso de registro y control de Actividades Complementarias después del desarrollo y ejecución de la aplicación web.	81

4.3.8.	Cuadro Comparativo.....	84
4.4.	Especificación de Requerimientos	86
4.4.1.	Introducción.....	86
4.4.2.	Propósito	86
4.4.3.	Alcance	87
4.4.4.	Definiciones, Acrónimos y Abreviaturas	88
4.4.5.	Referencias	89
4.5.	Descripción General.....	90
4.5.1.	Perspectiva del producto.....	90
4.5.2.	Funciones del producto	90
4.5.2.1.	Iniciar sesión.....	91
4.5.2.2.	Datos Iniciales	91
4.5.2.3.	Facultades y Carreras	93
4.5.2.4.	Períodos Lectivos	93
4.5.2.5.	Horarios de Docentes	93
4.5.2.6.	Informe de Actividades	93
4.5.2.7.	Horario de Actividades.....	93
4.5.3.	Características de los Usuarios.....	94
4.5.3.1.	Administradores del sistema:.....	94
4.5.3.2.	Docentes.....	94
4.5.3.3.	Autoridades.....	94
4.6.	Requerimientos Específicos.....	95
4.6.1.	Requerimientos Funcionales.....	95
4.6.1.1.	Casos de Uso	95
4.6.2.	Requerimientos No Funcionales	116
4.6.3.	Requerimientos del sistema	116
4.6.4.	Requerimientos de interfaz de usuario.....	116
4.6.5.	Requerimientos de Seguridad.....	116
4.6.6.	Especificaciones Técnicas	117
4.6.6.1.	Estructura del modelo de datos:	123
4.6.6.2.	Diagrama de Secuencia	130
4.6.6.3.	Diagrama de Entidad - Relación.....	133
4.6.6.4.	Diagrama de Actividades	135
4.6.6.5.	Diagrama de Clases	136

4.7.	Interfaz de Usuario	137
4.7.1.	Páginas Web Inicial y de Acceso al Sistema.....	137
4.7.2.	Página Web del Menú del Administrador	138
4.7.2.1.	Página Web de Mantenimiento de Tipos de Identidades.....	140
4.7.2.2.	Página Web de Mantenimiento de Tipos de Relación Laboral	141
4.7.2.3.	Página Web de Mantenimiento de Tipos de Tiempo Dedicado	141
4.7.2.4.	Página Web de Mantenimiento de Roles.....	142
4.7.2.5.	Página Web de Mantenimiento de Usuarios.....	142
4.7.2.6.	Página Web de Asignación de Roles a Usuarios.....	143
4.7.2.7.	Página Web de Mantenimiento de Tipos de Actividades	143
4.7.2.8.	Página Web de Asignación de Tipos de Actividades.....	144
4.7.2.9.	Página Web de Mantenimiento de Facultades	144
4.7.2.10.	Página Web de Mantenimiento de Carreras	145
4.7.2.11.	Página Web de Mantenimiento de Períodos Lectivos	145
4.7.3.	Página Web del Menú Docente.....	146
4.7.3.1.	Página Web de Mantenimiento de Horario de Actividades...	146
4.7.3.2.	Página Web de Mantenimiento Informes de Actividades.....	146
4.7.3.3.	Página Web de Consulta y Reporte de Horas Cumplidas en actividades Complementarias	147
4.7.4.	Página Web del Menú Coordinador	148
4.7.4.1.	Página Web de Aprobación de Horarios a Docentes.....	148
4.7.4.2.	Página Web de Aprobación de Informes de Actividades	148
4.7.4.3.	Página Web de Reporte de Horas de Docentes	149
4.7.5.	Test de satisfacción en el uso del software.....	150
4.7.5.1.	Resumen del test de satisfacción en el uso del software	150
4.7.5.2.	Permanencia del test por módulo	151
4.7.5.3.	Grado de aprobación por módulo	152
CAPITULO V.....		154
5.1.	Conclusiones	155
5.2.	Recomendaciones	157
CAPITULO VI.....		159
6.1.	Bibliografía.....	160
CAPITULO VII.....		163

7.1.	Manual de Operación.....	164
7.1.1.	Introducción.....	164
	FORMULARIO	DE
	ENCUESTA.....	177

INDICE DE ILUSTRACIONES

Ilustración 1. Flujo de Procesos RAD	9
Ilustración 2. Patrón Modelo Vista Controlador	16
Ilustración 3. Modelos basados en Entity Framework.....	19
Ilustración 4. Modelo de datos con Entity Framework.....	24
Ilustración 5. Acceso a datos mediante Arquitectura Entity Framework	28
Ilustración 6. Tecnologías agrupadas bajo el concepto Ajax	33
Ilustración 7. Aplicación web clásica sincrónica.....	34
Ilustración 8. Aplicación web Asincrónica (Ajax)	34
Ilustración 9. Asp.Net MVC y Entity Framework	60
Ilustración 10. Funciones del Administrador	90
Ilustración 11. Funciones de los Docentes	91
Ilustración 12. Funciones de las Autoridades	91
Ilustración 13. Caso de Uso - Iniciar sesión	95
Ilustración 14. Casos de Uso - Mantenimiento Datos Iniciales	97
Ilustración 15. Casos de Uso - Horarios e Informes.....	109
Ilustración 16. Diagrama Arquitectura 4 Capas	117
Ilustración 17. Diagrama de Secuencia - Crear Horario.....	130
Ilustración 18. Diagrama de Secuencia - Crear Período.....	131
Ilustración 19. Diagrama de Secuencia - Crear Informe	132
Ilustración 20. Diagrama de Base de Datos.....	134
Ilustración 21. Diagrama de Actividades.....	135
Ilustración 22. Diagrama de Clases	136
Ilustración 23. Página web inicial	137
Ilustración 24. Pantalla de inicio de sesión	137
Ilustración 25. Pantalla modal selección tipo de rol	138
Ilustración 26. Opciones del menú de datos iniciales	139
Ilustración 27. Opciones del menú de facultades y carreras.....	139
Ilustración 28. Opciones del menú de períodos lectivos	140
Ilustración 29. Mantenimiento de tipos de identificación	140
Ilustración 30. Mantenimiento de tipos de relación laboral	141
Ilustración 31. Mantenimiento de tipos de dedicación de tiempo	141
Ilustración 32. Mantenimiento de tipos de roles	142

Ilustración 33. Mantenimiento de usuarios.....	142
Ilustración 34. Asignación de roles a usuarios	143
Ilustración 35. Mantenimiento de tipos de actividades.....	143
Ilustración 36. Asignación de tipos de actividades.....	144
Ilustración 37. Mantenimiento de Facultades.....	144
Ilustración 38. Mantenimiento de datos de carreras	145
Ilustración 39. Mantenimiento de períodos lectivos	145
Ilustración 40. Mantenimiento de horario de actividades	146
Ilustración 41. Mantenimiento de informe de actividades	147
Ilustración 42. Consulta y reporte de horas cumplidas	147
Ilustración 43. Aprobación de horario de actividades.....	148
Ilustración 44. Aprobación informe de actividades.....	149
Ilustración 45. Consulta y reporte de horas aplicadas	149

INDICE DE GRÁFICOS

Gráfico 1. Satisfacción de sistema para informes de actividades complementarias	66
Gráfico 2. Procesos que generan más problemas en el registro y control de actividades complementarias.....	66
Gráfico 3. Aplicación web para optimizar el registro y control de actividades complementarias.	67
Gráfico 4. Uso de aplicación web para el registro y control de actividades complementarias.	68
Gráfico 5. Emisión rápida de reportes de informes de actividades complementarias.	68
Gráfico 6. Reducción del consumo de papelería usando aplicación web. ..	69
Gráfico 7. Ventajas de la disponibilidad de las aplicaciones web	70

INDICE DE CUADROS

Cuadro 1. Población: Docentes y Autoridades de la UTEQ (2015).....	58
Cuadro 2. Promedio mensual del tiempo usado con el proceso anterior.....	79
Cuadro 3. Costo del consumo mensual de recursos materiales con el proceso anterior.....	79
Cuadro 4. Requisitos e inconvenientes en el registro y control usando el proceso anterior.....	81
Cuadro 5. Promedio mensual de tiempo usando la aplicación web para el registro y control de actividades complementarias.....	82
Cuadro 6. Costo del consumo mensual de recursos materiales usando la aplicación web para el registro y control de actividades complementarias.....	82
Cuadro 7. Requisitos e inconvenientes en el registro y control usando la aplicación web para el registro y control de actividades complementarias.....	83
Cuadro 8. Comparativo entre el proceso anterior y después del desarrollo y ejecución de la aplicación web.....	84
Cuadro 9. Caso de uso - Iniciar Sesión.....	96
Cuadro 10. Caso de uso - Crear Tipos de Identificación.....	98
Cuadro 11. Caso de uso - Crear Relación Laboral.....	99
Cuadro 12. Caso de uso - Crear Tipo de Dedicación de Tiempo.....	100
Cuadro 13. Caso de uso - Crear Roles.....	101
Cuadro 14. Caso de uso - Crear Usuarios.....	102
Cuadro 15. Caso de uso - Asignar Roles a Usuarios.....	103
Cuadro 16. Caso de uso - Crear Tipos de Actividades.....	104
Cuadro 17. Caso de uso - Asignar Actividades a Docentes.....	105
Cuadro 18. Caso de uso - Crear Datos de Facultades.....	106
Cuadro 19. Caso de uso - Crear Datos de Carreras.....	107
Cuadro 20. Caso de uso - Crear Períodos.....	108
Cuadro 21. Caso de uso - Crear Horario de Actividades.....	110
Cuadro 22. Caso de uso - Crear Informe de Actividades.....	111
Cuadro 23. Caso de uso - Reporte de Horas Cumplidas.....	112
Cuadro 24. Caso de uso - Aprobar Horario de Actividades.....	113
Cuadro 25. Caso de uso - Aprobar Informe de Actividades.....	114
Cuadro 26. Caso de uso - Reporte de Horas en Informes de Actividades.....	115

Cuadro 27. Estructura Entidad Facultad	123
Cuadro 28. Estructura Entidad Carrera.....	123
Cuadro 29. Estructura Entidad Tipolden	124
Cuadro 30. Estructura Entidad RelaLabo	124
Cuadro 31. Estructura Entidad Dedicacion	124
Cuadro 32. Estructura Entidad Rol	124
Cuadro 33. Estructura Entidad Usuario	125
Cuadro 34. Estructura Entidad UsuarioRol	125
Cuadro 35. Estructura Entidad TipoActividad	125
Cuadro 36. Estructura Entidad Periodo	126
Cuadro 37. Estructura Entidad ActividadDocente	126
Cuadro 38. Estructura Entidad Horario	126
Cuadro 39. Estructura Entidad DetalleHorario	128
Cuadro 40. Estructura Entidad MaestroFormato.....	128
Cuadro 41. Estructura Entidad DetalleInforme.....	128
Cuadro 42. Estructura Entidad MaestroInforme.....	129
Cuadro 43. Estructura Entidad DetalleFormato	129
Cuadro 44. Resumen test de satisfacción en el uso del software.....	150
Cuadro 45. Tiempo de permanencia en el test por módulo	151
Cuadro 46. Grado de aprobación por módulo.....	152

RESUMEN EJECUTIVO

El proyecto se ha aplicado en el área administrativa de la Universidad Técnica Estatal de Quevedo, con el fin de desarrollar una aplicación web para el control de la gestión de las actividades complementarias realizadas por los docentes de la institución. Esta aplicación optimizará los procesos de registro y control de estas gestiones, permitirá el uso de la información de una forma compartida y segura, garantizando la disponibilidad de los datos en repositorios electrónicos altamente confiables, lo cual reduce en gran cantidad el uso de papel.

Para lograr este objetivo se ha utilizado la Metodología RAD por sus siglas en inglés (Rapid Application Development). La Metodología de Desarrollo de Aplicaciones Rápidas es una de las metodologías ágiles que más se usan en el medio, complementada con el modelo de desarrollo de sistemas Code First (Primero el Código), de igual forma para definir el alcance y propósito se aplicaron las normas IEEE830 como punto inicial en la estructuración de este proyecto.

La Tecnología empleada en este proyecto es totalmente actualizada, basándose principalmente en el uso de Visual Studio 2013, aplicando la herramienta de desarrollo C#, utilizando un ambiente de edición de código totalmente amigable en la creación de aplicaciones web, permitiendo desarrollar interfaces de usuarios muy funcionales y agradables. Para garantizar la seguridad y disponibilidad de los datos se utilizó el sistema de gestión de base de datos MS SQL Server 2012, mismo que cuenta con una excelente arquitectura y robustez en el tratamiento de datos electrónicamente.

Una vez desarrollada la aplicación web, comprobaremos la optimización y eficacia en los procesos de registros y control de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo, agilizando la gestión, ahorrando tiempo y dinero, y también protegiendo al medio ambiente al evitar el uso inapropiado de papelería.

EXECUTIVE SUMMARY

The project has been implemented in the administrative area of the State Technical University of Quevedo, to develop a web application to control the management of complementary activities undertaken by teachers of the institution. This application will optimize processes registration and control of these activities will allow the use of information shared and safely, ensuring the availability of highly reliable electronic data repositories, which greatly reduces paper usage amount.

To achieve this objective we have used the RAD methodology for its acronym in English (Rapid Application Development). Methodology Rapid Application Development is one of the most agile methodologies used in the medium, supplemented with Model Code First development of systems (First the Code), similarly to define the scope and purpose of the rules applied IEEE830 as a starting point in structuring this project.

The technology used in this project is completely updated, based primarily on the use of Visual Studio 2013, using the development tool C#, using an editing environment totally friendly code to create web applications, allowing develop interfaces very functional users and pleasant. The management system based MS SQL Server 2012, the same data that has an excellent architecture and robustness in electronic data processing was used to ensure the security and availability of data.

Once developed the web application, check the optimization and process efficiency records and control of the follow teachers of the State Technical University of Quevedo, streamlining management, saving time and money, and protecting the environment by avoiding the inappropriate use stationery.

CAPITULO I
MARCO CONTEXTUAL

1.1. Introducción

La Universidad Técnica Estatal de Quevedo, a través de los Subdecanos y Decanos de cada una de las facultades, elaboran el Distributivo Académico donde se especifican las horas clases de los docentes y sus respectivas actividades complementarias a fin de que cumplan con las cuarenta horas de carga horaria.

Se consideran Actividades Complementarias aquellas labores que se realizan fuera del horario normal de clases, estas pueden ser de los siguientes tipos: Coordinación de Proyecto Integrador, Coordinación del Área del Conocimiento, Coordinación de Carrera, Coordinación de Prácticas Pre Profesionales, Coordinación de Vinculación con la Colectividad, Coordinación de Proyectos de Investigación, Comisión CONSEP, Coordinador de Maestría, entre otras.

Dichas actividades se presentan planificadas en el Distributivo Académico que es enviado a la Comisión Académica de la universidad para su aprobación y después se realiza su respectiva revisión por parte del Consejo Académico y Vicerrectorado.

En este documento se detallan, por cada facultad y carrera, los docentes y su correspondiente número total de horas que deben cumplir en la realización de las actividades complementarias.

De acuerdo con lo establecido en los distributivos, los docentes deben informar a sus respectivos Coordinadores de Carrera el avance y cumplimiento de estas actividades mediante informes en formatos previamente establecidos por la Unidad de Planeamiento Académico; mismos que deben ser compilados por carrera para ser entregados a los decanos de cada una de las facultades quienes finalmente evalúan la información y la envían a la Unidad de Planeamiento Académico y al departamento de Talento Humano, donde se realizarán los respectivos controles del cumplimiento de las cargas horarias de los docentes.

Los informes reflejan además del cumplimiento de los docentes, las labores que realizan los coordinadores en bien de la educación y en bien de la

institución, ejerciendo dentro de sus Atribuciones y Responsabilidades, el compromiso de: “Realizar el seguimiento y evaluación académica del programa carrera de grado y postgrado, dictado por la facultad”, de acuerdo con el Estatuto Orgánico de Gestión Organizacional por Procesos de la Universidad Técnica Estatal de Quevedo.

1.2. Objetivos

1.2.1. General

Desarrollar la aplicación web para el control de la gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo.

1.2.2. Específicos:

- Efectuar el análisis del registro y control de la gestión de las actividades complementarias de los docentes de la UTEQ
- Elaborar el estudio del esquema de la aplicación web con la norma IEEE830
- Elaborar un sistema informático empleando herramientas de cuarta generación.

1.3. Hipótesis

La aplicación web optimizará los procesos de registros, control y gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo.

CAPITULO II
MARCO TEORICO

2.1. Fundamentación Teórica

2.1.1. Metodologías Ágiles en el Desarrollo de Software

(Canós, Letelier y Penadés, 2015). Según los autores de este artículo, 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.

(Canós, Letelier y Penadés, 2015). Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto.

2.1.1.1. Metodología Kanban

(Kniberg y Skarin, 2010). Kanban se basa en una idea muy simple: el trabajo en curso (Work In Progress, WIP) que debería limitarse, y sólo debería empezarse con algo nuevo cuando un bloque de trabajo anterior haya sido entregado o ha pasado a otra función posterior de la cadena.

(Kniberg y Skarin, 2010). El Kanban (o tarjeta señalizador) implica que se genera una señal visual para indicar que hay nuevos bloques de trabajo que pueden ser comenzados porque el trabajo en curso actual no alcanza el máximo acordado. Esto no suena muy revolucionario ni parece que vaya a afectar profundamente el rendimiento, cultura, capacidad y madurez del equipo y de la organización que les rodea. ¡Lo increíble es que sí lo hace! Kanban parece un cambio muy pequeño pero aun así cambia todos los aspectos de una empresa.

2.1.1.2. Metodología Scrum

(Kniberg y Skarin, 2010). Scrum se basa en iteraciones de tiempo fijo. Se puede elegir la duración de la iteración, pero la idea general es mantener la misma longitud de la iteración durante un período de tiempo, determinando así una cadencia.

- (Kniberg y Skarin, 2010). Inicio de la iteración: Se crea un plan de iteración, es decir, el equipo saca un número específico de elementos de la pila de producto, en base a las prioridades del dueño de producto y a cuánto piensa el equipo que puede terminar en una iteración.
- (Kniberg y Skarin, 2010). Durante la iteración: El equipo se centra en completar los elementos a los que se comprometió. Se fija el alcance de la iteración.
- (Kniberg y Skarin, 2010). Final de la iteración: El equipo muestra el código funcionando a las partes interesadas, idealmente este código debe ser potencialmente entregable (es decir, probado y listo para llevar). Entonces, el equipo hace una retrospectiva para discutir y mejorar su proceso.

2.1.1.3. Metodología XP (Xtreme Programming)

(Kniberg, 2007). 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.

(Kniberg, 2007). 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.

2.1.1.4. Metodología RAD (Rapid Application Development)

(Laboratorio Nacional de Calidad del Software, 2009). El desarrollo rápido de aplicaciones es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. El término fue usado originalmente para describir dicha metodología. La metodología de Martin implicaba desarrollo iterativo y la construcción de prototipos. Más recientemente, el término y su acrónimo se están usando en un sentido genérico, más amplio, que abarca una variedad de técnicas dirigidas al desarrollo de aplicaciones rápidas.

(Laboratorio Nacional de Calidad del Software, 2009). La metodología de desarrollo rápido de aplicaciones (RAD) se desarrolló para responder a la necesidad de entregar sistemas muy rápido. El enfoque de RAD no es apropiado para todos los proyectos. El alcance, el tamaño y las circunstancias, todo ello determina el éxito de un enfoque RAD.

(Laboratorio Nacional de Calidad del Software, 2009). El método RAD tiene una lista de tareas y una estructura de desglose de trabajo diseñada para la rapidez. El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering), tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y rapidez de ejecución. A continuación, se muestra un flujo de proceso posible para el desarrollo rápido de aplicaciones:

Ilustración 1. Flujo de Procesos RAD



Fuente: Laboratorio Nacional de Calidad del Software, 2009

(Laboratorio Nacional de Calidad del Software, 2009). RAD requiere el uso interactivo de técnicas estructuradas y prototipos para definir los requisitos de usuario y diseñar el sistema final. Usando técnicas estructuradas, el desarrollador primero construye modelos de datos y modelos de procesos de negocio preliminares de los requisitos. Los prototipos ayudan entonces al analista y los usuarios a verificar tales requisitos y a refinar formalmente los modelos de datos y procesos.

(Laboratorio Nacional de Calidad del Software, 2009). El ciclo de modelos resulta a la larga en una combinación de requisitos de negocio y una declaración de diseño técnico para ser usado en la construcción de nuevos sistemas. Los enfoques RAD pueden implicar compromisos en funcionalidad y rendimiento a cambio de permitir el desarrollo más rápido y facilitando el mantenimiento de la aplicación. Las principales ventajas que puede aportar este tipo de desarrollo son las siguientes:

- (Laboratorio Nacional de Calidad del Software, 2009). Velocidad de desarrollo.
- (Laboratorio Nacional de Calidad del Software, 2009). Calidad: según lo definido por el RAD, es el grado al cual un uso entregado resuelve las necesidades de usuarios así como el grado al cual un sistema entregado tiene costes de mantenimiento bajos. El RAD aumenta la calidad con la implicación del usuario en las etapas del análisis y del diseño.
- (Laboratorio Nacional de Calidad del Software, 2009). Visibilidad temprana debido al uso de técnicas de prototipado.
- (Laboratorio Nacional de Calidad del Software, 2009). Mayor flexibilidad que otros modelos.
- (Laboratorio Nacional de Calidad del Software, 2009). Ciclos de desarrollo más cortos.

(Laboratorio Nacional de Calidad del Software, 2009). Entre los principales inconvenientes que se pueden encontrar en el uso del desarrollo rápido de aplicaciones constan los siguientes:

- (Laboratorio Nacional de Calidad del Software, 2009). Características reducidas.
- (Laboratorio Nacional de Calidad del Software, 2009). Escalabilidad reducida.
- (Laboratorio Nacional de Calidad del Software, 2009). Más difícil de evaluar el progreso porque no hay hitos clásicos.

(Laboratorio Nacional de Calidad del Software, 2009). Una de las críticas principales que suele generar este tipo de desarrollo es que, el desarrollo rápido de aplicaciones es un proceso iterativo e incremental, y puede conducir a una sucesión de prototipos que nunca culmine en una aplicación de producción satisfactoria. Tales fallos pueden ser evitados si las herramientas de desarrollo de la aplicación son robustas, flexibles y colocadas para el uso correcto.

(Laboratorio Nacional de Calidad del Software, 2009). Prácticas Ágiles: Son varias las prácticas que se utilizan en el desarrollo rápido. Aunque algunas de ellas se consideran metodologías en sí mismas, son simplemente prácticas usadas en diferentes metodologías.

(Laboratorio Nacional de Calidad del Software, 2009). Test Driven Development (TDD): El desarrollo orientado a pruebas (TDD) es una técnica de desarrollo de software que usa iteraciones de desarrollo cortas basadas en casos de prueba escritos previamente que definen las mejoras deseadas o nuevas funcionalidades. Cada iteración produce el código necesario para pasar la prueba de la iteración. Finalmente, el programador o equipo refactoriza el código para acomodar los cambios.

(Laboratorio Nacional de Calidad del Software, 2009). Un concepto clave de TDD es que las pruebas se escriben antes de que se escriba el código para que éste cumpla con las pruebas. Hay que tener en cuenta que TDD es un método de diseño de software, no sólo un método de pruebas. TDD está relacionado con los primeros conceptos de pruebas de programación de Extreme Programming, en 1999, pero más recientemente se está creando un interés general mayor en sí mismo.

(Laboratorio Nacional de Calidad del Software, 2009). TDD requiere que los desarrolladores creen pruebas unitarias automatizadas para definir los requisitos del código antes de escribir el código en sí mismo. Las pruebas contienen afirmaciones que son verdaderas o falsas. El objetivo es escribir código claro que funcione. Ejecutar las pruebas rápidamente confirma el comportamiento correcto a medida que los desarrolladores evolucionan y refactorizan el código.

(Laboratorio Nacional de Calidad del Software, 2009). Los desarrolladores se ayudan de herramientas para crear y ejecutar automáticamente conjuntos de casos de prueba. Una ventaja de esta forma de programación es evitar escribir código innecesario. Se intenta escribir el mínimo código posible. La idea de escribir las pruebas antes que el código tiene dos beneficios principales. Ayuda a asegurar que la aplicación se escribe para poder ser probada, porque los desarrolladores deben considerar cómo probar la aplicación desde el principio, en vez de preocuparse por ello luego. También asegura que se escriben pruebas para cada característica.

2.1.2. Aplicaciones web

Según (Carles, 2014). Inicialmente la web era simplemente una colección de páginas estáticas, documentos, etc., que podían consultarse o descargarse.

(Carles, 2014). El éxito espectacular de la web se basa en dos puntales fundamentales: el protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones que permite enviar cualquier tipo de ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones y reduzcan los costes de despliegue. El otro proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple.

(Carles, 2014). El protocolo HTTP (hypertext transfer protocol) es el protocolo base de la www. Se trata de un protocolo simple, orientado a conexión y sin estado. La razón de que esté orientado a conexión es que emplea para su funcionamiento un protocolo de comunicaciones (TCP, transport control

protocol) de modo conectado, un protocolo que establece un canal de comunicaciones de extremo a extremo (entre el cliente y el servidor) por el que pasa el flujo de bytes que constituyen los datos que hay que transferir, en contraposición a los protocolos de datagrama o no orientados a conexión que dividen los datos en pequeños paquetes (datagramas) y los envían, pudiendo llegar por vías diferentes del servidor al cliente. El protocolo no mantiene estado, es decir, cada transferencia de datos es una conexión independiente de la anterior, sin relación alguna entre ellas.

((OMG), 2015). Para representar el funcionamiento de este sistema, se utilizará el lenguaje de modelado UML por sus siglas en inglés (Unified Modeling Language), mismo que permitirá describir el sistema en sus diferentes etapas y componentes, donde se exponen detalles como procesos de negocios, modelo de datos, modelo de componentes, entre otros esquemas en los cuales se apoya el proceso de desarrollo de esta aplicación utilizando metodologías orientadas a objetos, siendo necesaria la creación de los siguientes diagramas:

2.1.3. Diagrama de casos de uso

Según (Kendall, 2011). Un modelo de caso de uso describe lo que hace un sistema sin describir como lo hace; es decir, es un modelo lógico del sistema. El modelo de caso de uso refleja la vista del sistema desde la perspectiva de un usuario fuera del sistema.

Este diagrama permite documentar el comportamiento de un sistema en una forma simple y fácil de interpretar, el cual se corresponderá con su respectivo diagrama de Especificación de Casos de Uso.

2.1.4. Diagrama de actividades

Según (Kendall, 2011). Los diagramas de actividades muestran las secuencias de actividades de un proceso, incluyendo las actividades secuenciales, las actividades paralelas y las decisiones que se toman.

Mediante este diagrama, se expondrá el flujo de trabajo identificado en las especificaciones de los casos de uso.

2.1.5. Diagrama de secuencias

Según (Kendall, 2011). Los diagramas de secuencias ilustran una sucesión de interacciones entre clases o instancias de objetos en un período determinado. Los diagramas de secuencia se utilizan con frecuencia para representar el proceso descrito en los escenarios de casos de uso.

Este tipo de diagrama expone la interacción entre un conjunto de objetos a través del tiempo, el cual se modela para cada caso de uso, conteniendo detalles de la implementación.

2.1.6. Diagrama de clases

Según (Kendall, 2011). Los diagramas de clases muestran las características estáticas del sistema y no representan ningún procesamiento en particular. Un diagrama de clase también muestra la naturaleza de las relaciones entre las clases.

En los diagramas de clases se muestra la arquitectura y estructura del sistema y sus respectivos subsistemas, asegurando el diseño de la aplicación, descubriendo las clases, atributos, métodos y relaciones.

2.1.7. Lenguaje de programación C#

(MSDN, 2015). Esta aplicación se desarrolló utilizando el lenguaje de programación C# versión 5.0, esta herramienta pertenece a la suite de VISUAL STUDIO 2013 y por su total compatibilidad con SQL SERVER 2012 se utilizará este sistema de gestión de bases de datos, contando ambas tecnologías con el respaldo tecnológico de Microsoft Corporation, asegurando así la integridad, seguridad y disponibilidad de la información, previendo que en el futuro se trataría con un gran volumen de datos.

2.1.8. Programación Orientada a Objetos (POO).

La metodología de desarrollo a utilizarse es la orientada a objetos (POO).

Según (Joyanes, 1998). La programación orientada a objetos es un importante conjunto de técnicas que pueden utilizarse para hacer el desarrollo de

programas más eficientes, a la par que mejora la fiabilidad de los programas de computadora. En la programación orientada a objetos, los objetos son los elementos principales de construcción. Sin embargo, la simple comprensión de lo que es un objeto, o bien el uso de objetos en un programa, no significa que esté programado en un modo orientado a objetos. Lo que cuenta es el sistema en el cual los objetos se interconectan y se comunican entre sí.

Este tipo de metodología permite aplicar conceptos de la vida real en la creación de estos sistemas, así como también la reusabilidad y extensibilidad del código, para obtener una aplicación más limpia y más fácil de distribuir y mantener.

2.1.9. Arquitectura ASP.NET

(MSDN, 2015). El patrón de diseño está bajo la arquitectura ASP.NET utilizando el esquema MVC versión 5 por su siglas en inglés (MODEL VIEW CONTROLLER), el cual permite desarrollar aplicaciones web mucho más versátiles que las aplicaciones web ASP.NET tradicionales, además esta tecnología ya incluye excelentes herramientas como BOOTSTRAP CSS, JQUERY, AJAX, que permiten aplicar estilos y temas a las páginas web dotadas de código JAVA SCRIPT y controles AJAX mediante los cuales se puede crear aplicaciones de tipo sensibles (RESPONSIVE) de acuerdo al dispositivo que las utilice, y también aplicaciones con un mejor rendimiento en cuanto a los procesos de respuesta por parte del servidor.

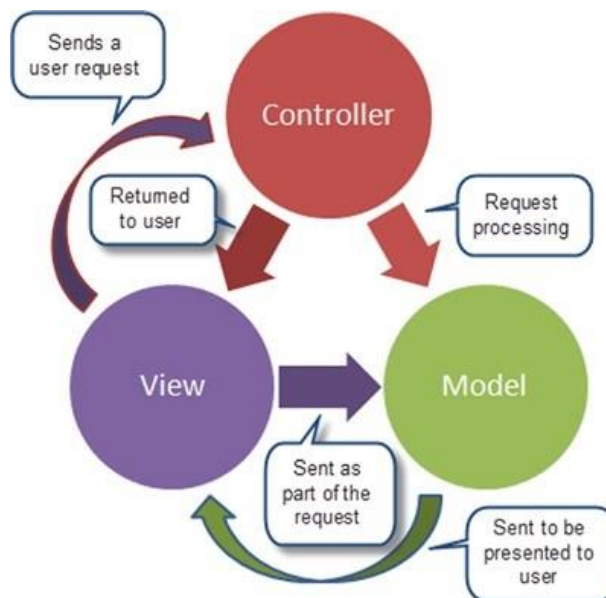
(Guay, 2013). ASP.NET MVC es un marco de trabajo libre y de apoyo para la creación de aplicaciones web que utilizan el patrón modelo-vista-controlador. Al igual que los formularios Web ASP.NET, ASP.NET MVC se construye en la parte superior del marco ASP.NET. Esto significa que se puede utilizar en aplicaciones ASP.NET MVC las mismas APIs para la seguridad, la administración del estado, la pertenencia, el almacenamiento en caché, etc., como las que se utilizan en aplicaciones Web Forms ASP.NET tradicionales.

En el mundo de ASP.NET MVC, muchas mejoras en ASP.NET se han incluido en el propio marco. El propósito principal de este patrón de diseño es aislar la lógica de negocio de la interfaz de usuario con el fin de centrarse en una mejor

capacidad de mantenimiento, la mejora de la capacidad de prueba, y una estructura más limpia a la aplicación. Cada aplicación ASP.NET MVC tiene tres partes fundamentales: un modelo, vistas y controladores. En resumen, el modelo consta de todas las clases que se ocupan de los datos y la lógica de negocio. Los Procesamientos de datos utilizando las clases del modelo son iniciados por los controladores que están a cargo de las peticiones del usuario. Una vez que el procesamiento de datos se ha completado el controlador crea una respuesta al usuario mediante el envío de los resultados a una vista que luego produce HTML a ser visualizados en el navegador.

2.1.10. Modelo vista controlador (MVC)

Ilustración 2. Patrón Modelo Vista Controlador



Fuente: Microsoft, MSDN, 2015

(MSDN, 2015). El modelo en el patrón MVC representa las partes de la aplicación que implementa la lógica de dominio de datos, las flechas curvadas indican las asociaciones indirectas, mientras que las flechas rectas indican las asociaciones directas, las vistas (View) son los elementos visibles en la aplicación, estos son los componentes que muestran a los usuarios los controles y los datos de los modelos (Model), una página de vista típicamente recibe un objeto de vista de modelo del controlador (Controller), esta página de

vista contiene código HTML y posiblemente código de instrucciones como javascript, código de algún motor de vista (ASPX, RAZOR).

(MSDN, 2015). Los controladores son clases que recogen las peticiones de los usuarios a través de las vistas, trabajan con los modelos aplicando ciertas reglas de negocio, obteniendo datos, y presentar una interfaz apropiada nuevamente en una vista. Microsoft Office Project 2013 es la herramienta que se utilizará para la gestión de este proyecto, con el cual se administrará eficientemente y se llevará un correcto seguimiento del trabajo, optimizando tiempo y recursos.

(Guay, 2013). Las aplicaciones ASP.NET MVC no solo se benefician mediante la inclusión de todas las características principales de ASP.NET, sino también por las características en el patrón MVC. Algunos de estos beneficios son: El patrón MVC en sí hace que sea más fácil de manejar la complejidad, separando claramente la funcionalidad de la aplicación en tres partes fundamentales, el modelo, la vista y el controlador.

(Guay, 2013). Las Aplicaciones web ASP.NET MVC no utilizan el estado de vista o formas basadas en servidor. Esto hace el marco idóneo MVC para los desarrolladores que desean tener un control total sobre el comportamiento de una aplicación. ASP.NET MVC funciona bien en aplicaciones web que son apoyados por grandes equipos de desarrolladores y para diseñadores web que necesitan un alto grado de control sobre el HTML.

2.1.11. Modelo CodeFirst Basado en Entity Framework

(Lerman y Miller, 2011). La primera versión de Entity Framework, que vino como parte de .NET 3.5 y Visual Studio 2008, permitió a los desarrolladores la capacidad de crear este tipo de modelo conceptual de la ingeniería inversa de una base de datos existente en un archivo XML. Este archivo XML utiliza la extensión EDMX, y se podía utilizar un diseñador para ver y personalizar el modelo para adaptarse mejor a su dominio. Visual Studio 2010 y .NET 4 trajeron la segunda versión de Entity Framework, llamada EntityFramework 4 (EF4), para alinearse con la versión .NET. En el lado de modelado, se agregó una nueva función denominada Primero el Modelo, aquí se puede diseñar el

modelo conceptual visualmente y luego crear la base de datos basado en el modelo.

(Lerman y Miller, 2011). ModelFirst permite a los desarrolladores que trabajan en nuevos proyectos que no cuentan con bases de datos heredadas, de beneficiarse de las prestaciones de Entity Framework. También los desarrolladores pueden comenzar con un enfoque en su dominio de aplicación mediante el diseño del modelo conceptual y dejar que el flujo de creación de base de datos realice ese proceso.

(Lerman y Miller, 2011). Diseñando el archivo EDMX en base al modelo DATABASE FIRST o MODEL FIRST, el siguiente paso para la creación del dominio es la generación automática de las clases cuyo código se basa en las entidades y sus relaciones que encuentra en el modelo. A partir de aquí, los desarrolladores pueden utilizar clases fuertemente tipadas en representación de sus objetos de dominio, pudiendo acceder con total comodidad a los nombres de las entidades y sus propiedades.

a. El Inicio de CodeFirst:

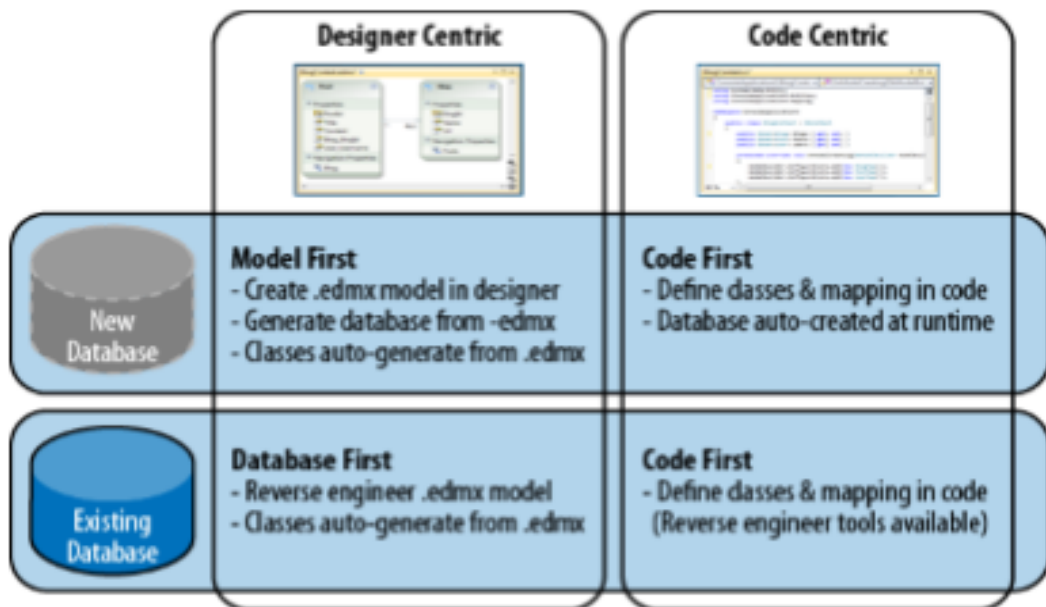
(Lerman y Miller, 2011). Basándose en las características que se introdujeron en EF4, Microsoft fue capaz de crear un camino más para el modelado, que muchos desarrolladores han estado solicitando desde inicios del EF. Este nuevo tipo de modelo se llama CodeFirst. CodeFirst permite definir el modelo de dominio con código en lugar de utilizar un archivo EDMX basado en XML. Aunque ModelFirst y DatabaseFirst de la primera generación de código se utiliza para trabajar con clases, muchos desarrolladores simplemente no quieren trabajar con un diseñador ni tener sus clases generadas por ellos. Sólo querían escribir código.

(Lerman y Miller, 2011). En CODE FIRST se comienza por la definición de su modelo de dominio utilizando clases POCO, que no tienen dependencia de Entity Framework. CodeFirst puede inferir una gran cantidad de información acerca de su modelo puramente de la forma de sus clases. También puede proporcionar una configuración adicional para describir con más detalle el

modelo o anular lo que CodeFirst infiere. Esta configuración también se define en el código: no hay archivos XML o diseñadores.

(Lerman y Miller, 2011). CodeFirst, DataBaseFirst, y ModelFirst son todas formas de la construcción de un modelo de desarrollo basado en entidades de datos que se pueden utilizar con Entity Framework para realizar el acceso de datos. Una vez que el modelo ha sido construido, el tiempo de ejecución de Entity Framework se comporta con la misma independencia de cómo se creó el modelo.

Ilustración 3. Modelos basados en Entity Framework



Fuente: Programming Entity Framework, Lerman y Miller, 2011

(Lerman y Miller, 2011). Microsoft recomienda el uso de la clase DbContext para trabajar con CodeFirst, debido a que estas proporcionan un acceso más sencillo a los patrones que son más complejas de lograr con la claseObjectContext. DbContext también se ocupa de muchas tareas comunes, por lo que se escribe menos código para lograr las mismas tareas.

b. Manipulando Objetos con DbContext:

(Lerman y Miller, 2011). Las clases de dominio descritas anteriormente no tienen nada que ver con el Entity Framework. No tienen conocimiento de él. Eso es lo bueno de trabajar con CodeFirst. Se tiene la oportunidad de utilizar

sus propias clases. Esto es especialmente beneficioso si se tiene clases de dominio existentes de un proyecto anterior. Para utilizar CodeFirst, se empieza por la definición de una clase que hereda de la clase DbContext. Una de las funciones de esta clase, que nos referiremos a ella como contexto, es dejar que CodeFirst sepa acerca de las clases que componen el modelo. Así es como Entity Framework será consciente de ello y será capaz de realizar un seguimiento de estas clases.

Esto se realiza mediante la exposición de las clases de dominio a través de otra nueva clase introducida junto con DbContext, la clase DbSet. Así como DbContext es un envoltorio sencillo alrededor delObjectContext, DbSet es una envoltura ObjectSet alrededor de Entity Framework, lo que simplifica las tareas de codificación para los que utilizan normalmente el ObjectSet.

c. Uso de la capa de datos y las clases de dominio:

(Lerman y Miller, 2011). A continuación se realiza un proceso que puede parecer un poco sorprendente. Esto es todo lo que se necesita para una capa de datos que se basa en la suposición de que usted va a confiar al 100 por ciento al modelo de desarrollo CodeFirst, quien a continuación en base a las convenciones hará el resto del trabajo. Aún no existe una cadena de conexión de base de datos. No hay ni siquiera una base de datos. Pero aun así, el proceso está listo para usar esta capa de datos.

(Lerman y Miller, 2011). En tiempo de ejecución, Entity Framework lee el XML que describe las tres partes del XML y crea una representación en memoria de los metadatos. Pero los metadatos en memoria no son XML; son objetos de clases fuertemente tipados como EntityType, EdmProperty y AssociationType. Entity Framework interactúa con esta representación en memoria del modelo y esquema cada vez que necesita interactuar con la base de datos.

(Lerman y Miller, 2011). Debido a que no hay un archivo XML con CodeFirst, este crea los metadatos en memoria de los archivos que encuentra en sus clases de dominio. Aquí es donde la convención y la configuración entran en juego. CodeFirst tiene una clase llamada DbModelBuilder. Esta clase lee las clases del dominio y construye el modelo en memoria a la medida de sus

posibilidades. Dado que también es la construcción de la parte de los metadatos que representa el esquema de base de datos, que es capaz de utilizar para crear la base de datos. Se pueden agregar configuraciones para ayudar al constructor de modelos a determinar cuál es el modelo y el esquema de la base.

d. Trabajando con configuraciones:

(Lerman y Miller, 2011). En aquellos casos en que CodeFirst necesita alguna ayuda para entender su propósito desarrollando bajo este modelo, existen dos opciones para realizar la configuración: Data Annotations y Fluent API. La opción que escoja se basa más a menudo en las preferencias personales y su estilo de codificación. Hay un poco de configuración avanzada que sólo es posible a través de la FluentAPI. CodeFirst le permite configurar una gran variedad de atributos de propiedades, relaciones, jerarquías de herencia y las asignaciones de bases de datos. Para este efecto deberá respetar las opciones de convenciones y de configuración que están disponibles.

e. Configurando con Data Annotations:

(Lerman y Miller, 2011). Una forma de aplicar la configuración, aprovechada por muchos desarrolladores por su simplicidad, es el uso de Data Annotations. Data Annotations son atributos que se aplican directamente a la clase o propiedades que desea modificar. Estos se pueden encontrar en la librería o espacio de nombres `System.ComponentModel.DataAnnotations`. Por ejemplo, si usted quiere asegurarse de que una propiedad debe tener siempre un valor, puede utilizar la anotación "Required".

(Lerman y Miller, 2011). Esto tendrá dos efectos. La primera es que el campo `TypeName` en la base de datos se convertirá en una columna que no acepta valores nulos. El segundo es que será validado por Entity Framework, gracias a la API de validación que también se introdujo en Entity Framework. Por defecto, cuando se realice el proceso de salvar los cambios (`SaveChanges`), Entity Framework verificará el campo para asegurarse de que la propiedad que ha

marcado como Requerido no esté vacía. Si está vacío, Entity Framework lanzará una excepción.

(Lerman y Miller, 2011). La anotación Required afecta las actualizaciones de la columna de base de datos y validación de la propiedad. Algunas anotaciones son específicas sólo para las asignaciones de bases de datos. Por ejemplo, las anotaciones sobre la Tabla indican a CodeFirst que mapee la clase a una tabla de un cierto nombre. Por ejemplo los datos que se refieren como AnimalType en una aplicación pueden ser almacenado en una tabla llamada Species.

f. Configurando con el Fluent API:

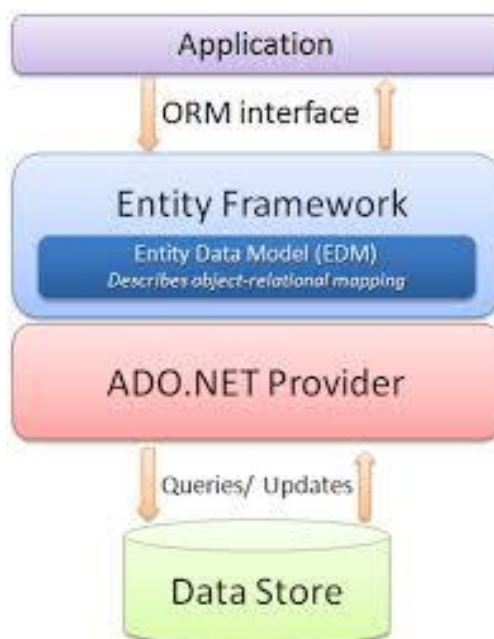
(Lerman y Miller, 2011). Aplicar las configuraciones con Data Annotations es muy simple de hacer, especificando los metadatos dentro de una clase de dominio no puede alinearse con su estilo de desarrollo. Existe una forma alternativa para agregar configuraciones que utiliza la Fluent API de CodeFirst. Con la configuración aplicada a través de Fluent API, sus clases de dominio siguen siendo "limpia." En lugar de modificar las clases, se proporciona la información de configuración del generador de modelos de CodeFirst en un método expuesto por la clase DbContext. El método se llama OnModelCreating.

(Lerman y Miller, 2011). La primera configuración utiliza el equivalente de Fluent API de Data Annotations de tabla, que es el método ToTable, y pasa en el nombre de la tabla a qué clase de tipo de entidad debe ser mapeada. La segunda configuración utiliza una expresión lambda para identificar una de las propiedades del tipo de entidad y luego añade el método IsRequired a esa propiedad. Esta es sólo una manera de construir configuraciones Fluent API. Se pueden hacer mucho más con el uso tanto de Data Annotations como con Fluent API para configurar atributos de propiedades, relaciones, jerarquías de herencia y las asignaciones de base de datos.

(Lerman y Miller, 2011). Los modelos de desarrollo CodeFirst, ModelFirst y DatabaseFirst basados en Entity Framework han evolucionado a través de las mejoras que ha tenido este marco de trabajo (Entity Framework), Entity Framework no es solo una herramienta enfocada en las distintas bases de

datos que hay en el mercado y que han desarrollado los drivers para este marco de trabajo. Entity Framework en base a sus potentes herramientas permite desarrollar las clases (POCOs) Plain Old CLR Objects, permitiendo un desarrollo de software de dominio centrado y sin privar de sus derechos a los desarrolladores sobre los datos de entrada. Se mejoraron características como el rendimiento y numerosos detalles sobre la calidad del código generado.

Ilustración 4. Modelo de datos con Entity Framework



Fuente: Programming Entity Framework, Lerman y Miller, 2011

(Lerman y Miller, 2011). Una importante mejora es que ahora el modelo de desarrollo CodeFirst admite la asignación a procedimientos almacenados en cualquier base de datos que haya desarrollado sus drivers para este marco de trabajo, este avance es algo que ha sido apoyado por los modelos creados en el diseñador. Esta característica ha recibido mucha cobertura en Canal 9 vídeos (como el de bit.ly/16wL8fz) y en una especificación detallada en el sitio de intercambio de conocimientos CodePlex.

(Lerman y Miller, 2011). Entity Framework ya no es un marco de trabajo de Visual Studio, este ha sido liberado como una extensión de Visual Studio. Para EF6, que puede ser instalado a través de los Nuget Package. Esto es una gran ventaja. De cara al futuro, el equipo será capaz de añadir características directamente al diseñador, incluyendo las que se facilitan en las herramientas de Entity Framework.

(Lerman y Miller, 2011). CodeFirst utiliza la característica de Entity Framework en la reutilización de conexiones abiertas, así como también utiliza la gran característica llamada Resiliency Connection. Si una conexión se interrumpe mientras EF está intentando ejecutar una consulta o guardando los cambios,

ahora tiene la capacidad de indicarle a EF que reintente el proceso. Esta característica es muy útil para ayudar a las aplicaciones que se conectan a la nube. Los reintentos se pueden configurar mediante `IDbConnectionStrategy`, de igual forma la clase `SqlAzureExecutionStrategy`, está configurada para las conexiones a base de datos de Windows Azure SQL.

(Lerman y Miller, 2011). Una característica muy importante del modelo de desarrollo CodeFirst basado en Entity Framework, es la capacidad para crear scripts de migración que pueden comprobar los cambios realizados en el dominio de clases y aplicar estos cambios en el modelo físico de datos actualizando la estructura de la base de datos, guardando un registro de estos scripts por fecha y hora, que incluso pueden ser personalizados antes de ejecutarlos.

2.1.12. Generalidades de Entity Framework

(Lerman, 2013). Entity Framework es un conjunto de tecnologías de ADO.NET que permiten el desarrollo de aplicaciones de software orientadas a datos. Los arquitectos y programadores de aplicaciones orientadas a datos se han enfrentado a la necesidad de lograr dos objetivos muy diferentes. Deben modelar las entidades, las relaciones y la lógica de los problemas empresariales que resuelven, y también deben trabajar con los motores de datos que se usan para almacenar y recuperar los datos. Los datos pueden abarcar varios sistemas de almacenamiento, cada uno con sus propios protocolos; incluso las aplicaciones que funcionan con un único sistema de almacenamiento deben equilibrar los requisitos del sistema de almacenamiento con respecto a los requisitos de escribir un código de aplicación eficaz y fácil de mantener.

(Lerman, 2013). Entity Framework permite a los desarrolladores trabajar con datos en forma de objetos y propiedades específicos del dominio, como clientes y direcciones de cliente, sin tener que preocuparse por las tablas y columnas de la base de datos subyacente donde se almacenan estos datos. Con Entity Framework, los desarrolladores pueden trabajar en un nivel mayor de abstracción cuando tratan con datos, y pueden crear y mantener

aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales.

(Lerman, 2013). Entity Framework da vida a los modelos conceptuales permitiendo a los programadores consultar las entidades y relaciones en el modelo de dominio (denominado modelo conceptual en Entity Framework) al tiempo que se basan en Entity Framework para traducir esas operaciones en los comandos específicos del origen de datos. Esto libera a las aplicaciones de las dependencias codificadas de forma rígida en un origen de datos determinado. Al trabajar con el modelo de desarrollo CodeFirst, el modelo conceptual se asigna al modelo de almacenamiento en código.

(Lerman, 2013). Entity Framework puede deducir el modelo conceptual basado en los tipos de objeto y configuraciones adicionales que define. Los metadatos de asignación que se generan durante el tiempo de ejecución basándose en una combinación de cómo se definen los tipos de dominio e información de configuración adicional que se proporciona en código. Entity Framework genera la base de datos como sea necesario basándose en los metadatos.

(Lerman, 2013). Las herramientas de Entity Data Model generan clases de datos extensibles según el modelo conceptual. Se trata de clases parciales que se pueden extender con miembros adicionales que el programador agrega. De forma predeterminada, las clases que se generan para un modelo conceptual determinado derivan de las clases base que proporcionan servicios para materializar las entidades como objetos y para realizar un seguimiento de los cambios y guardarlos. Los desarrolladores pueden utilizar estas clases para trabajar con las entidades y relaciones como objetos relacionados mediante asociaciones. Los desarrolladores también pueden personalizar las clases que se generan para un modelo conceptual.

(Lerman, 2013). Como algo más que otra solución de asignación objeto-relacional, Entity Framework trata fundamentalmente de permitir que las aplicaciones obtengan acceso y cambien los datos que están representados como entidades y relaciones en el modelo conceptual. Entity Framework usa la información de los archivos del modelo y de asignación para traducir las

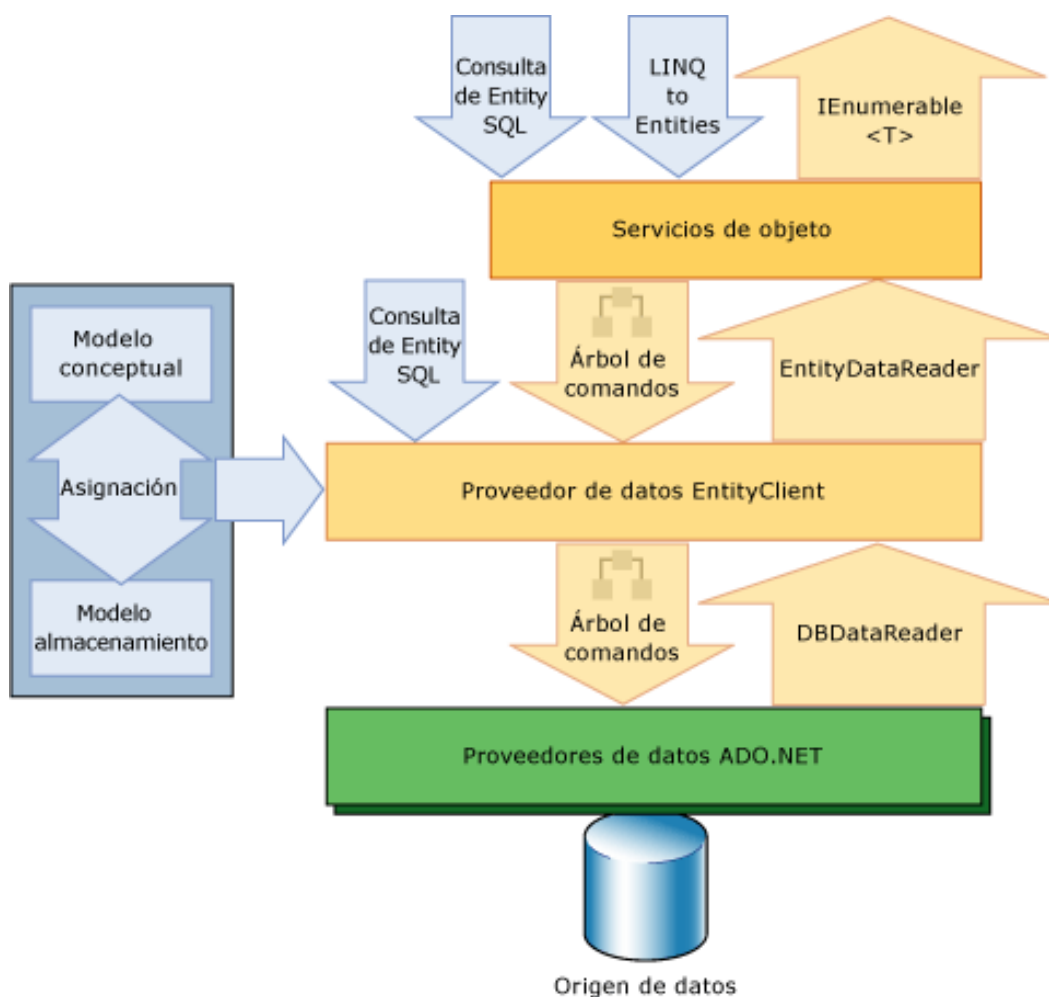
consultas de objeto con los tipos de entidad que se representan en el modelo conceptual en consultas específicas del origen de datos. Los resultados de la consulta se materializan en objetos que Entity Framework administra. Entity Framework proporciona las siguientes maneras de consultar un modelo conceptual y devolver objetos:

- (Lerman, 2013). LINQ to Entities. Proporciona compatibilidad con Language-Integrated Query (LINQ) para consultar los tipos de entidad que se definen en un modelo conceptual.
- (Lerman, 2013). Entity SQL. Un dialecto de SQL, independiente del almacenamiento, que trabaja directamente con entidades del modelo conceptual y que admite conceptos de Entity Data Model. Entity SQL se utiliza tanto con consultas de objeto como con consultas que se ejecutan utilizando el proveedor EntityClient.

(Lerman, 2013). El Entity Framework incluye el proveedor de datos de EntityClient. Este proveedor administra las conexiones, traduce las consultas de entidad en consultas específicas del origen de datos y devuelve un lector de datos que Entity Framework usa para materializar los datos de la entidad en los objetos. Cuando no se requiere la materialización de los objetos, el proveedor de EntityClient también se puede utilizar como un proveedor de datos ADO.NET estándar habilitando las aplicaciones para ejecutar consultas Entity SQL y usar el lector de datos de solo lectura devuelto.

En la siguiente figura se muestra un esquema del acceso a datos aplicando la arquitectura Entity Framework, y basada en consultas del tipo Entity SQL y LINQ to Entities a través del servicio de objetos, el cual aplica los proveedores de datos (EntityClient) y por medio de los componentes de ADO.NET realiza la interacción entre el modelo conceptual y el modelo de almacenamiento:

Ilustración 5. Acceso a datos mediante Arquitectura Entity Framework



Fuente: MSDN Magazine, Lerman, 2013

2.1.13. Microsoft Project

Según (Chatfield y Johnson, 2013). Microsoft Project 2013 es una herramienta poderosa para la creación y gestión de proyectos. Es una potente aplicación que te ayuda a planificar y gestionar una amplia gama de proyectos. Desde el cumplimiento de plazos y presupuestos cruciales hasta la selección de los recursos adecuados, puedes ser más productivo y entregar mejores resultados utilizando el conjunto de características de Microsoft Project 2013.

2.1.14. Motor Razor

(Chadwick, 2011). Razor es una de las sintaxis de ASP.NET MVC utilizadas en plantillas que permiten combinar código y contenido de una manera fluida y expresiva. A pesar de que utiliza algunos símbolos y palabras clave, Razor no

es un nuevo lenguaje. Razor permite escribir código utilizando lenguajes como C # o Visual Basic .NET dentro de una vista combinándose con el Html.

(Chadwick, 2011). El símbolo @ es el corazón de la sintaxis Razor, este es el personaje que Razor utiliza para diferenciar el código de marcado. El símbolo @ marca un punto en el que el desarrollador tiene la intención de cambiar de marcado con el código. En casos sencillos, no se necesitan caracteres adicionales para indicar cuándo el código termina. El analizador inteligente determina qué partes de la plantilla son de código Razor y que son de códigos Html.

Sentencia if/else usando sintaxis Web Forms:

```
<% if(User.IsAuthenticated) { %>
<span>Hello, <%= User.Username %>!</span>
<% } %>
<% else{ %>
<span>Please<%= Html.ActionLink("log in") %></span>
<% } %>
```

Sentencia if/else usando sintaxis Razor:

```
@if(User.IsAuthenticated)
{ <span>Hello, @User.Username!</span> }
else
{ <span>Please @Html.ActionLink("log in")</span> }
```

2.1.15. JavaScript

(Haverbeke, 2011). JavaScript es el lenguaje más utilizado por el momento, aplicado en su mayoría para hacer todo tipo de cosas inteligentes y horribles en páginas web en la World Wide Web. Algunas personas afirman que la próxima versión de JavaScript se convertirá en un idioma importante para otras tareas también. No es seguro que eso pase, pero para las personas interesadas en la programación, JavaScript es definitivamente un lenguaje útil para aprender. Incluso si no se termina haciendo mucha programación web.

(Crockford, 2008). JavaScript es un lenguaje importante porque es el idioma del navegador web. Su asociación con el navegador hace que sea uno de los lenguajes de programación más populares en el mundo. Es uno de los lenguajes de programación web que interactúa con la API del navegador, el Document Object Model (DOM).

(Eguiluz, 2009). Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems.

Posibilidades y Limitaciones

(Eguiluz, 2009). Desde su aparición, JavaScript siempre fue utilizado de forma masiva por la mayoría de sitios de Internet. La aparición de Flash disminuyó su popularidad, porque Flash permitía realizar algunas acciones imposibles de llevar a cabo mediante JavaScript. Sin embargo, la aparición de las aplicaciones AJAX programadas con JavaScript le ha devuelto una popularidad sin igual dentro de los lenguajes de programación web.

(Eguiluz, 2009). En cuanto a las limitaciones, JavaScript fue diseñado de forma que se ejecutara en un entorno muy limitado que permitiera a los usuarios confiar en la ejecución de los scripts. De esta forma, los scripts de JavaScript no pueden comunicarse con recursos que no pertenezcan al mismo dominio desde el que se descargó el script.

Los scripts tampoco pueden cerrar ventanas que no hayan abierto esos mismos scripts. Las ventanas que se crean no pueden ser demasiado pequeñas ni demasiado grandes ni colocarse fuera de la vista del usuario (aunque los detalles concretos dependen de cada navegador). Además, los scripts no pueden acceder a los archivos del ordenador del usuario (ni en modo lectura ni en modo escritura) y tampoco pueden leer o modificar las preferencias del navegador. Por último, si la ejecución de un script dura

demasiado tiempo (por ejemplo por un error de programación) el navegador informa al usuario de que un script está consumiendo demasiados recursos y le da la posibilidad de detener su ejecución.

2.1.16. JQuery

(Chaffer, 2013). Hoy en día la worldwide web (www) es un entorno dinámico y los desarrolladores de aplicaciones web encuentran un gran obstáculo tanto para el estilo y la función de los sitios. Para construir sitios interesantes e interactivos, los desarrolladores están recurriendo a bibliotecas JavaScript como jQuery para automatizar tareas comunes y simplificar las complicadas. Una de las razones de la biblioteca jQuery que es una opción popular es su capacidad para ayudar en una amplia gama de tareas.

(Chaffer, 2013). La librería jQuery proporciona una capa de abstracción de propósito general para scripting web común, y por lo tanto es útil en casi todas las situaciones de scripting. Su naturaleza extensible significa que nunca podríamos cubrir todos los posibles usos y funciones en un solo libro, como constantemente se están desarrollando plugins para añadir nuevas capacidades. Las características fundamentales, sin embargo, ayudan a llevar a cabo las siguientes tareas:

- (Chaffer, 2013). Elementos de acceso en un documento: Sin una biblioteca de JavaScript, los desarrolladores web a menudo tienen que escribir muchas líneas de código para atravesar el Document Object Model (DOM) y localizar partes específicas de la estructura de un documento HTML. Con jQuery, los desarrolladores tienen un mecanismo selector robusto y eficiente a su disposición, por lo que es fácil de recuperar la pieza exacta del documento que debe ser inspeccionado o manipulado. `$('.div.content').find('p');`
- (Chaffer, 2013). Modificar la apariencia de una página web: CSS ofrece un potente método de influir en la forma en que un documento se hace, pero se queda corto cuando no todos los navegadores soportan los mismos estándares. Con jQuery, los desarrolladores pueden llenar este vacío, confiando en los mismos estándares de apoyar en todos los

navegadores, además jQuery puede cambiar las clases o propiedades de estilo individuales aplicadas a una parte del documento, incluso después de que la página ha sido presentada. `$('#ul> li: first').addClass("activo");`

- (Chaffer, 2013). Altera el contenido de un documento: No se limita a meros cambios cosméticos, jQuery puede modificar el contenido de un documento propio con sólo pulsar unas teclas. El texto puede ser cambiado, las imágenes pueden insertarse o intercambiarse, las listas pueden ser reordenadas, o la totalidad de la estructura del HTML pueden ser reescritos y ampliados. `$('# contenedor').append(' más ');`
- (Chaffer, 2013). Animar cambios hechos a un documento: Para aplicar efectivamente tales comportamientos interactivos, un diseñador también debe proporcionar información visual al usuario. La librería jQuery facilita esto proporcionando una serie de efectos como fundidos y toallitas húmedas, así como un conjunto de herramientas para la elaboración de otros nuevos. `$('.div.detailsslideDown');`
- (Chaffer, 2013). Recuperar información de un servidor sin actualizar una página: Este patrón de código se conoce como el Ajax, que originalmente se encontraba de Asynchronous JavaScript y XML, pero desde entonces ha llegado a representar un mayor conjunto de tecnologías para la comunicación entre el cliente y el servidor. La librería jQuery elimina la complejidad específica del navegador de este proceso de respuesta, rica en características, permitiendo a los desarrolladores centrarse en la funcionalidad de servidor.

2.1.17. Ajax

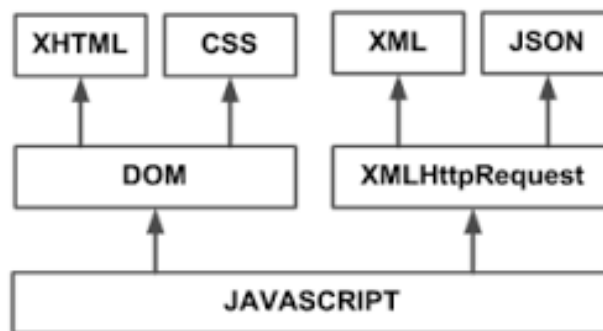
(Eguiluz, 2008). El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo. En realidad, el término AJAX es un acrónimo de

Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML".

(Eguiluz, 2008). Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Ilustración 6. Tecnologías agrupadas bajo el concepto Ajax



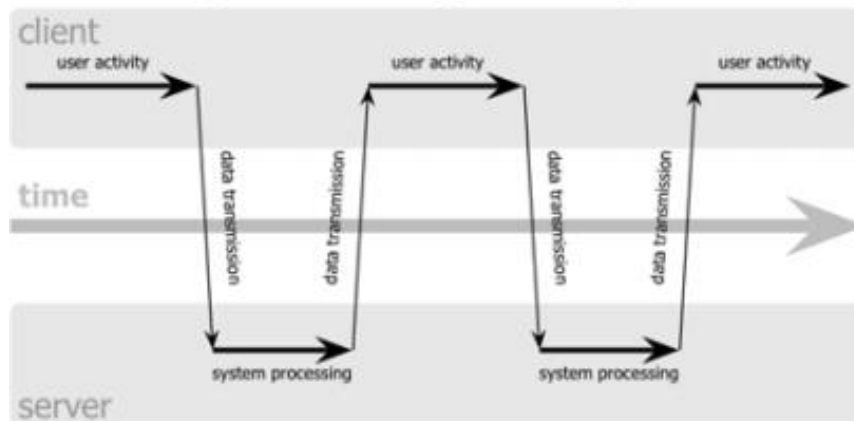
Fuente: Introducción a Ajax, Eguiluz, 2008

(Eguiluz, 2008). Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, así el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

(Eguiluz, 2008). El siguiente esquema muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX. Las imágenes inferiores muestran la comunicación sincrónica y asíncrona de las aplicaciones creadas sin AJAX y con AJAX.

Ilustración 7. Aplicación web clásica sincrónica

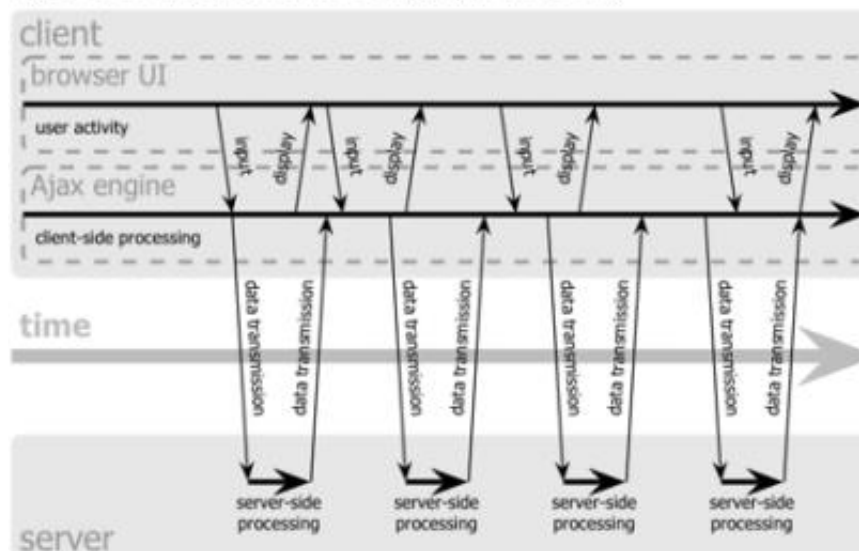
classic web application model (synchronous)



Fuente: **Introducción a Ajax, Eguiluz, 2008**

Ilustración 8. Aplicación web Asíncrona (Ajax)

Ajax web application model (asynchronous)



Fuente: **Introducción a Ajax, Eguiluz, 2008**

(Eguiluz, 2008). Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones

más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

2.1.18. Bootstrap 3.0

(Wikipedia, 2012). Twitter Bootstrap es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. Bootstrap fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un marco de trabajo (framework) para fomentar la consistencia a través de herramientas internas, fue liberado por twitter como código abierto en Febrero del 2012.

Características, Estructuras y Función:

(Wikipedia, 2012). Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

2.1.19. HTML5

(Gauchat, 2012). HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

(Gauchat, 2012). Conceptos básicos: HTML (LENGUAJE DE MARCAS DE HIPERTEXTO) usa un lenguaje de etiquetas para construir páginas web. Estas etiquetas HTML son palabras clave y atributos rodeados de los signos mayor y menor (por ejemplo, <html lang="es">). En este caso, html es la palabra clave y lang es el atributo con el valor es. La mayoría de las etiquetas HTML se utilizan en pares, una etiqueta de apertura y una de cierre, y el contenido se declara entre ellas. En nuestro ejemplo, <html lang="es"> indica el comienzo del código HTML y </html> indica el final. Compare las etiquetas de apertura y cierre y verá que la de cierre se distingue por una barra invertida antes de la palabra clave (por ejemplo, </html>). El resto de nuestro código será insertado entre estas dos etiquetas: <html> ...</html>, el código a continuación nos muestra la estructura básica de un documento HTML5:

```
<!DOCTYPEhtml>

  <html lang="es">
  <head>
    <meta charset="iso-8859-1" />
    <meta name="description" content="Ejemplo de HTML5" />
    <meta name="keywords" content="HTML5, CSS3, JavaScript" />
  </head>
  <body>
  </body>
</html>
```

(Gauchat, 2012). HTML5 es, de hecho, una mejora de todas las versiones anteriores a fin de soportar tecnologías como JQuery, JavaScript, Bootstrap CSS3. HTML5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. A partir de ahora, HTML5 provee los elementos estructurales, CSS se encuentra concentrado en cómo volver esa estructura utilizable y atractiva a la vista, y Javascript tiene todo el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales.

(Gauchat, 2012). HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript provee la funcionalidad necesaria que permite la interacción del usuario con las aplicaciones web, dotadas de excelentes herramientas como JQuery y Ajax.

2.1.20. Visual Studio

(Microsoft Corporation, 2014). Visual Studio 2013 es la suite de Microsoft con poderosas herramientas para la planificación, modelado, desarrollo y testeo de aplicaciones de escritorio, aplicaciones web, servicios web y aplicaciones móviles, posee en su suite los lenguajes de programación como Visual Basic, Visual C#, Visual C++, F# y una enriquecida interfaz de desarrollo (IDE), con excelentes controles y opciones que facilitan y simplifican el desarrollo, depuración, instalación, publicación de aplicaciones informáticas, la versión 2013 de Visual Studio integra el marco de trabajo (FRAMEWORK) versión 4.5.1, cuya actualización a la fecha es la Update-3, la cual es una librería completa que implementa todos los componentes necesarios para la creación de aplicaciones de cualquiera de los tipos anteriormente mencionados.

(Jeff, 2014). La llegada de Visual Studio 2013 (VS2013) marca la continuación del deseo de Microsoft para producir cambios más frecuentes a la plataforma Visual Studio. Trabajando en VS2013, se encuentra un aspecto más familiar. Los usuarios que actualicen a VS2013 desde versiones anteriores a VS2012 encontrarán mayores diferencias y tendrán que invertir un poco más de tiempo en el aprendizaje de la nueva interfaz.

(Jeff, 2014). VS2013 se ha centrado en gran medida en la mejora de la usabilidad y el poder del editor utilizado en el desarrollo de aplicaciones independientemente de si el usuario escribe aplicaciones de consola de C++ o

el uso de JavaScript para escribir aplicaciones de Windows Store. Visual Studio 2013 habilitará ciertas opciones disponibles según la versión de Windows que esté ejecutando. Por ejemplo si está ejecutando Windows 8.1, podrá crear nuevas aplicaciones de Windows Store para Windows 8.1.

2.1.21. Regular Expressions

(Fitzgerald, 2012). Las expresiones regulares son especialmente cadenas de texto codificados utilizados como patrones para emparejar juegos de caracteres. Comenzaron a surgir en la década de 1940 como una manera de describir lenguajes regulares, pero en realidad comenzaron a aparecer en el mundo de la programación durante la década de 1970. Utilizados en el editor de texto QED escrito por Ken Thompson. Las expresiones regulares más tarde se convirtieron en una parte importante de la suite de herramientas del sistema operativo Unix-la ed, sed y vi (vim) editores, grep, awk, entre otros.

(Friedl, 2012). Las expresiones regulares son herramientas poderosas, de gran alcance y flexibilidad en procesamiento de textos. Las expresiones regulares son consideradas casi como un lenguaje de programación, permiten describir y analizar el texto. Con el apoyo adicional proporcionado por una herramienta particular que se utilice, las expresiones regulares pueden añadir, eliminar, aislar, e incluso mutilar todo tipo de texto.

(Friedl, 2012). NET Framework de Microsoft, utilizado con Visual Basic, C# y C++ (entre otros idiomas), ofrece una biblioteca compartida regular-expressions que unifica la semántica de expresiones regulares entre los lenguajes de programación. Es un potente motor con todas las funciones, que le permite la flexibilidad máxima en el desarrollo de aplicaciones que implementan este tipo de control y validación en la manipulación de datos.

Las expresiones regulares pueden ser usadas dentro del lenguaje de programación, así como también en una página Html que reconozca este tipo de atributo.

Este ejemplo comprueba si una expresión regular coincide con una cadena:

```
If Regex.IsMatch(TestStr , "\s+$")
```

```
        Console.WriteLine("line is empty");
Else
        Console.WriteLine("line is not empty");
EndIf
```

Este ejemplo valida el correcto ingreso de un número telefónico con una expresión regular dentro del código Html.

```
<input id="telefono" placeholder="888-888-8888" title="XXX-XXX-XXXX"
required pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}">
```

2.1.22. JSon

(Howdle, 2012). JSON (JavaScript Object Notation) es una forma ligera, fácil y popular para intercambiar datos. jQuery no es la única herramienta para la manipulación y la interfaz, Json se ha convertido en una de las herramientas más preferidas para este tipo de trabajo. Una gran cantidad de los servicios que utilizamos todos los días tienen APIs basadas en JSON: Twitter, Facebook y Flickr todos envían datos en formato JSON.

(Howdle, 2012). La respuesta JSON es accesible a través de una función de jQuery, lo que permite manipular, visualizar y, más importante aún, el estilo que queramos. jQuery es la biblioteca JavaScript, con funciones útiles como getJSON, esta función en particular será la relación que mantiene junto la API cliente Json. Actualmente Json puede ser construido sobre dos estructuras: Object que es una colección de nombres y valores pares y Array que es una lista ordenada de valores como se muestra en los siguientes ejemplos.

Objeto con dos miembros:

```
{
    "Identificacion" : "1201889076",
    "nombre" : "Wilmer Cevallos"
}
```

Arreglo con tres elementos:

```
["Lacteo", "Yogur durazno", 85.00]
```

Uso de la API Json con JQuery:

```
$("#ReporPerID").change(function () {  
  
    var url = "@Url.Action("MesesDelPeriodo", "Reportes)";  
    var data = { periodoID: $(this).val() };  
    $.getJSON(url, data, function (result) {  
        var mesesAnios = $("#AnioMesCoordID");  
        mesesAnios.empty();  
        $(result).each(function () {  
            $(document.createElement('option'))  
                .attr('value', this.ID)  
                .text(this.AnioMes)  
                .appendTo(mesesAnios);  
        });  
    });  
});
```

2.1.23. DBMS

(Ramez y Shamkant, 2010). Las bases de datos y los sistemas de bases de datos (DBMS) son un componente esencial de la vida en la sociedad moderna: la mayoría de nosotros encontramos varias actividades todos los días que implican algún tipo de interacción con una base de datos. Por ejemplo, si vamos al banco a depositar o retirar fondos, si hacemos una reserva de hotel o línea aérea, si accedemos a un catálogo de una biblioteca informatizada para buscar un elemento bibliográfico, o si compramos algo en línea, tales como un libro, juguete, o de la computadora más probable es que nuestras actividades implicarán alguien o algún programa de ordenador para acceder a una base de datos. Incluso la compra de artículos en un supermercado actualiza automáticamente la base de datos que contiene el punto de venta.

(Ramez y Shamkant, 2010). Las bases de datos y la tecnología de base de datos tienen un impacto importante en el creciente uso de las computadoras, es justo decir que las bases de datos juegan un papel crítico en casi todas las

áreas donde se utilizan computadoras, incluidas las empresas, el comercio electrónico, la ingeniería, la medicina, la genética, el derecho, la educación y la bibliotecología. La palabra base de datos se usa con tanta frecuencia que hay que empezar por definir qué es una base de datos. Una base de datos es una colección de datos relacionados, se refiere a hechos conocidos que pueden ser registrados y que tienen en ellos sentidos implícitos. Por ejemplo, considere los nombres, números de teléfono y direcciones de las personas a las que haya registrado en una libreta de direcciones indexada o puede que haya almacenado en un disco duro, utilizando una computadora personal y software como Microsoft Access o Excel. Esta colección de datos relacionados con un significado implícito es una base de datos y podríamos decir que tiene las siguientes propiedades implícitas:

- (Ramez y Shamkant, 2010). Una base de datos representa algún aspecto del mundo real, a veces llamado el medio mundo o el universo, los cambios en el medio mundo se reflejan en la base de datos.
- (Ramez y Shamkant, 2010). Una base de datos es una colección lógica coherente de datos con algún significado inherente. Un surtido aleatorio de datos no puede correctamente ser referido como una base de datos.
- (Ramez y Shamkant, 2010). Una base de datos es diseñada, construida y poblada con datos para un propósito específico. Cuenta con un grupo de usuarios previstos y algunas aplicaciones preconcebidas en las que estos usuarios están interesados.

Según (Ramez y Shamkant, 2010). Un sistema de gestión de base de datos (DBMS) es un conjunto de programas que permiten a los usuarios crear y mantener una base de datos. El Sistema de Gestión de Bases de Datos es un software de propósito general que facilita los procesos de definición, construir, manipular y compartir bases de datos entre los distintos usuarios y aplicaciones. Definir una base de datos implica especificar los tipos de datos, estructuras, y las limitaciones de los datos que se almacena en ella, la información descriptiva también se almacena por el DBMS en la forma de un catálogo de base de datos o diccionario, se llama meta-data a la base de datos

resultante del proceso de almacenar los datos en algún medio de almacenamiento que está controlado por el DBMS.

(Ramez y Shamkant, 2010). **Ventajas del uso de Sistemas de Bases de Datos:**

- (Ramez y Shamkant, 2010). **Control de redundancia:** En el enfoque de base de datos, las opiniones de los diferentes grupos de usuarios se integran en el diseño de bases de datos. Idealmente, deberíamos tener un diseño de base de datos que almacena cada elemento lógico de datos tales como el nombre o la fecha de nacimiento de un estudiante en un solo lugar en la base de datos. Esto se conoce como la normalización de datos, y asegura la consistencia y ahorra espacio de almacenamiento. Sin embargo, en la práctica, a veces es necesario utilizar la redundancia controlada para mejorar el rendimiento de las consultas.
- (Ramez y Shamkant, 2010). **Restricción al acceso no autorizado:** Cuando varios usuarios comparten una gran base de datos, es probable que la mayoría de los usuarios no estarán autorizados a acceder a toda la información en la base de datos. Por ejemplo, los datos financieros es a menudo considerada confidencial, y sólo las personas autorizadas puedan acceder a dichos datos. Un DBMS debe proporcionar una seguridad y autorización a subsistemas, que el DBA utiliza para crear cuentas y especificar las restricciones, entonces el DBMS debe hacer cumplir estas restricciones de forma automática. Nótese que podemos aplicar controles similares al software DBMS. Por ejemplo, sólo el personal del DBA se puede permitir el uso de determinado software privilegiado, como el software para crear nuevas cuentas. Del mismo modo, los usuarios parametrizados pueden ser autorizados a acceder a la base de datos sólo a través de las transacciones desarrolladas para su uso.
- (Ramez y Shamkant, 2010). **Proveer almacenamiento persistente a objetos de programas:** El almacenamiento persistente de objetos de programa y estructuras de datos es una función importante de los sistemas de bases de datos. Sistemas de bases de datos tradicionales a menudo sufrían el problema de falta de concordancia así llamado,

porque las estructuras de datos proporcionados por el DBMS eran incompatibles con las estructuras de datos del lenguaje de programación. Sistemas de bases de datos orientadas a objetos suelen ofrecer estructura de datos compatibles con uno o más lenguajes de programación orientados a objetos.

- (Ramez y Shamkant, 2010). **Proporcionar estructuras de almacenamiento y técnicas de búsqueda para el procesamiento de consultas eficiente:** El DBMS a menudo tiene un módulo de almacenamiento en búfer o caché que mantiene partes de la base de datos en los principales buffers de memoria. En general, el sistema operativo es el responsable del almacenamiento en caché de disco a memoria, porque el almacenamiento en el búfer de datos es crucial para el rendimiento DBMS, la mayoría de los DBMS hacen su propio almacenamiento intermedio de datos. El módulo de procesamiento y optimización de consultas del DBMS es responsable de la elección de un plan de ejecución de la consulta eficiente para cada consulta basada en la estructura. El almacenamiento existente de los índices para crear y mantener es parte del diseño de base de datos física y puesta a punto, que es uno de las responsabilidades del DBA para discutir el proceso de la consulta, la optimización y puesta a punto.
- (Ramez y Shamkant, 2010). **Proporcionar respaldo y recuperación de datos:** Un DBMS debe proporcionar facilidades para la recuperación de fallos de hardware o software. La copia de seguridad y recuperación del subsistema DBMS es responsable de la recuperación. Por ejemplo, si el sistema informático falla en medio de una transacción de actualización complejo, el subsistema de recuperación es responsable de asegurarse de que la base de datos se restaura al estado en que se encontraba antes de la operación, alternativamente el subsistema de recuperación podría asegurar que la transacción se reanuda desde el punto en que se interrumpió de modo que su efecto total se registra en la base de datos. El respaldo en disco también es necesaria en caso de fallo catastrófico del disco

- (Ramez y Shamkant, 2010). **Proporcionar múltiples interfaces de usuarios:** Debido a que muchos tipos de usuarios con diferentes niveles de conocimientos técnicos utilizan una base de datos, un DBMS debe proporcionar una variedad de interfaces de usuario. Estos incluyen lenguajes de consulta para los usuarios ocasionales, interfaces de lenguaje de programación para programadores de aplicaciones, formas y códigos de comando para usuarios parametrizados e interfaces basadas en menús e interfaces de lenguaje natural para usuarios independientes. Ambas interfaces y formas de estilo y las interfaces basadas en menús son comúnmente conocidas como interfaces gráficas de usuario (GUI).
- (Ramez y Shamkant, 2010). **Hacer cumplir las restricciones de integridad:** La mayoría de las aplicaciones de bases de datos tienen ciertas restricciones de integridad que deben mantener para los datos. Un DBMS deberían tener capacidad para definir y hacer cumplir estas restricciones, el tipo más simple de restricción de integridad implica la especificación de un tipo de datos para cada elemento de los datos, también tenemos la integridad referencial que controla la manipulación de registros en tablas relacionadas.

2.1.24. RDBMS

Según (Ramez y Shamkant, 2010). El modelo relacional representa la base de datos como una colección de relaciones. Informalmente, cada relación se asemeja a una tabla de valores o, en cierta medida, un archivo plano de registros. Esto se llama un archivo plano, debido a que cada registro tiene una estructura plana sencilla, lineal. Sin embargo, hay diferencias importantes entre las relaciones y los archivos.

(Ramez y Shamkant, 2010). Cuando una relación se considera como una tabla de valores, cada fila de la tabla representa un conjunto de valores de datos relacionados. Una fila representa un hecho que normalmente corresponde a una entidad del mundo real o relación. Los nombres de las columnas y los nombres de las tablas se utilizan para ayudar a interpretar el significado de los valores de cada fila. Por ejemplo, si tenemos una tabla estudiante cada fila

representa hechos acerca de un estudiante, esta tabla tiene la columna Student_number que es utilizada en la relación de esta tabla con las otras entidades vinculadas debiendo interpretar los valores de los datos en cada fila, en base al valor de cada columna que deben ser del mismo tipo de datos. En la terminología modelo relacional formal, una fila se llama una tupla, un encabezado de columna se llama un atributo, y la tabla se llama una relación.

(Ramez y Shamkant, 2010). Especificar la restricción de integridad referencial entre dos relaciones utilizadas para mantener la consistencia entre tuplas en las dos relaciones. Informalmente, la restricción de integridad referencial establece que una tupla en una relación que se refiere a otra relación debe referirse a una tupla existente en esa relación.

2.1.25. SQL Server

(Microsoft, 2012). SQL Server 2012 la base de la amplia plataforma de datos de Microsoft, ofrece un rendimiento fiable gracias a la integración de tecnologías en memoria, una rápida obtención de información útil a partir de cualquier tipo de datos con herramientas como: Excel, Acces, entre otros, y una plataforma para compilar, implementar y administrar soluciones tanto locales como en nube.

(Microsoft, 2012). SQL Server 2012 permite obtener más rendimiento de los datos, poniendo a su disposición una amplia gama de servicios integrados como son consultas, búsquedas, sincronizaciones, informes y análisis. Sus datos pueden almacenarse y recuperarse desde sus servidores más potentes de un Data Center hasta los dispositivos móviles, logrando tener un mayor control sobre la información sin importar dónde se encuentre almacenada físicamente.

(Microsoft, 2012). SQL Server 2012 integra funcionalidad que, además de aportar fiabilidad, permite revelar información útil mediante el uso de herramientas de análisis conocidas y soluciones Big Data preparadas para la empresa. Su arquitectura y herramientas comunes para entornos locales y en nube hacen posibles las infraestructuras de TI híbridas. Incluye nueva funcionalidad en memoria en la base de datos principal y también proporciona

nuevas funciones preparadas para la nube que simplifican su adopción. SQL Server 2012 permite utilizar sus datos en aplicaciones desarrolladas con Visual Studio 2013 y también desde su propia Arquitectura Orientada a Servicio (SOA) y los procesos empresariales empleando Microsoft® BizTalk® Server.

(Microsoft, 2012). SQL Server 2012 es una plataforma escalable de Business Intelligence optimizada para la integración de datos, elaboración de informes y análisis que hace posible poner al alcance de todos los usuarios la inteligencia empresarial.

2.1.26. Common Language Runtime (CLR)

(Microsoft, 2015). A partir de SQL Server 2005, SQL Server incorpora la integración del componente Common Language Runtime (CLR) de Microsoft .NET Framework para Windows. Esto significa que ahora se pueden escribir procedimientos almacenados, desencadenadores, tipos definidos por el usuario, funciones definidas por el usuario, agregados definidos por el usuario, así como funciones con valores de tabla de transmisión por secuencias mediante cualquier lenguaje de .NET Framework, incluidos Microsoft Visual Basic .NET y Microsoft Visual C#.

(Microsoft, 2015). SQL Server 2012 es el motor de base de datos escalable y de alto rendimiento en requerimientos de alta concurrencia en aplicaciones de misión crítica con las mayores exigencias de disponibilidad y seguridad.

2.1.27. Data Warehouse

(Microsoft, 2015). SQL Server posee una plataforma de Data Warehouse completa y escalable que permite integrar datos dentro de esta tecnología más rápidamente, escalando y gestionando grandes volúmenes de datos facilitando el acceso a esta información desde cualquier herramienta ya sea externa o interna, como por ejemplo Reporting Services, optimizando el tiempo de entrega y análisis de datos a nivel operativo, gerencial y directivo, facilitando la toma de decisiones y planificación de las estrategias de negocio.

2.1.28. Norma IEEE

(Mendez, 2008). Según IEEE, un buen Documento de Requisitos, pese a no ser obligatorio que siga estrictamente la organización y los formatos dados en el estándar 830, sí deberá incluir, de una forma o de otra, toda la información presentada en dicho estándar.

Según (Mendez, 2008). El estándar de IEEE 830 no está libre de defectos ni de prejuicios, y por ello ha sido justamente criticado por múltiples autores y desde múltiples puntos de vista, llegándose a cuestionar incluso si es realmente un estándar en el sentido habitual que tiene el término en otras ingenierías.

Según (Mendez, 2008). Un Documento de Requisitos según el estándar IEEE 830 se organizaría de la siguiente forma:

(Mendez, 2008). **Introducción:** En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software (ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento, como se enumeran a continuación:

- Propósito
- Ámbito del sistema
- Definiciones, acrónimos y abreviaturas
- Referencias
- Visión general del documento

(Mendez, 2008). **Descripción General:** En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender. Normalmente, esta sección consta de las siguientes subsecciones:

- Perspectiva del producto
- Funciones del producto
- Características de los usuarios
- Restricciones

- Factores que se asumen y futuros requisitos.

(Mendez, 2008). **Requisitos Específicos:** Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS:

- Interfaces externas
- Funciones
- Requisitos del rendimiento
- Restricciones de diseño
- Atributos del sistema
- Otros Requisitos

(Mendez, 2008). **Apéndices:** Pueden contener todo tipo de información relevante para la ERS pero que, propiamente, no forme parte de la ERS. Por ejemplo:

- Formatos de entrada/salida de datos, por pantalla o en listados.
- Resultados de análisis de costes.
- Restricciones acerca del lenguaje de programación.

2.1.29. Actividades Complementarias

(UTEQ-UPA, 2015). Las Actividades Complementarias son trabajos realizados fuera del horario normal de clases, con el propósito de completar las cuarenta horas de carga horaria.

(UTEQ-UPA, 2015). Según resolución adaptada por el Honorable Consejo Universitario, Los Docentes dentro de su dedicación académica, no deberán tener más de tres comisiones generales como actividad complementaria.

2.1.30. UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO (UTEQ)

(UTEQ, 2015). **Misión**

Formar profesionales y académicos con visión científica y humanista capaces de desarrollar investigaciones, crear tecnologías, mantener y difundir nuestros saberes y culturas ancestrales, para la construcción de soluciones a los problemas de la región y el país.

(UTEQ, 2015). **Visión**

La Universidad Técnica Estatal de Quevedo ocupa los primeros lugares entre las instituciones de Educación Superior ecuatorianas por su calidad académica, investigativa y de servicios que contribuyen al Buen Vivir.

(UTEQ, 2015). **Historia**

La Universidad Técnica Estatal de Quevedo (UTEQ), se encuentra ubicada en Quevedo, ciudad central y capital económica de la Provincia de Los Ríos.

Esta prestigiosa institución se inició el 22 de enero de 1976, como Extensión Universitaria con la carrera de Ingeniería Forestal e Ingeniería Zootécnica, dependiente de la Facultad de Ciencias Agropecuarias de La Universidad "Luís Vargas Torres" de Esmeraldas.

Fue creada como Universidad Técnica Estatal de Quevedo mediante Ley de la República del 26 de enero de 1984, publicada en el Registro Oficial No. 674 del 1 de Febrero de 1984.

La Universidad nace con la creación de la Facultad de Ciencias Agropecuarias, y sus Escuelas de Ingeniería Forestal, Ingeniería Zootécnica, y las Tecnologías en Manejo de Suelo y Agua y Mecanización Agrícola; luego, con el transcurrir del tiempo se crea la Facultad de Zootecnia.

Posteriormente, son creadas las Escuelas de Ingeniería en Administración de Empresas Agropecuarias y las Tecnologías Agrícola, Agroindustrial y Topografía Agrícola.

Además de las anteriores, también se crean las escuelas de Computación, Banca y Finanzas, Ventas y Microempresas.

En la actualidad el nombre de estas últimas fueron modificadas, dando así la creación de la Facultad de Ciencias Empresariales con sus cuatro escuelas: Escuelas de Informática, Escuela de Economía y Finanzas, Escuela de Mercadotecnia y la Escuela de Gestión Empresarial.

Actualidad

Actualmente la institución cuenta con la Unidad de Estudios a Distancia (UED), el Centro de Idiomas Extranjeros (CEDI), la Unidad de Posgrado, un Instituto de Informática, una extensión universitaria en la ciudad de la Maná, provincia de Cotopaxi, y diversas oficinas de apoyo en varios cantones dentro y fuera de la provincia de Los Ríos, y es así que en los actuales momentos es la universidad pionera en la formación de profesionales, prestos al servicio de nuestro país.

CAPITULO III
METODOLOGIA DE LA INVESTIGACIÓN

3.1. Ubicación

Esta investigación se realizará en la Provincia de los Ríos, Cantón Quevedo, en la Universidad Técnica Estatal del cantón Quevedo, ubicada en la Parroquia Venus en la Avenida Quito Km. 1½ vía a Santo Domingo. El mismo que tendrá una duración de 6 meses, y está dirigido al personal de docentes, coordinadores de carrera y directivos de la Unidad de Planeamiento Académico (UPA) de la institución antes mencionada.

3.2. Materiales y métodos

3.2.1. Materiales y Equipos

RECURSOS HUMANOS	Cantidad
Investigación	5
RECURSOS TECNOLÓGICOS	
Computadoras	3
Memory Flash de 4 Gb	2
CDs	5
Carpetas	10
Marcadores	4
Perforadoras	1
Resma de papel	10
Xerox copias	410
SOFTWARE	
Sistema Operativo Windows 8	1
Sistema Operativo Windows Server 2008	1
Visual Studio 2013	1
Office Project 2013, Office Excel 2013, Office Word 2013	1

Internet Explorer, Google Chrome, Mozilla FireFox	1
SQL Server 2012	1

3.3. Método

Para esta investigación se aplicaron los siguientes métodos:

3.3.1. Inductivo- deductivo:

Permitió particularizar y generalizar los procesos que se realizan para el registro, control y gestión de las actividades complementarias de los docentes de la UTEQ.

3.3.2. Descriptivo:

Por medio de este método, se describe la situación real de los usuarios involucrados en este proceso, en cada una de las etapas y se determinaron sus funciones durante estas actividades.

3.3.3. Analítico o Explicativo:

Se plantea como un gran objetivo la optimización de los procesos de registro para el control de la gestión de las actividades complementarias de los docentes de la UTEQ, para esto se efectuó el análisis del gran cambio que implica la automatización de estas actividades en una forma más profesional y ordenada en beneficio de los propios usuarios de esta aplicación.

3.4. Tipo de Investigación

Para el desarrollo de un software o solución informática, es de primordial importancia comprender detalladamente el problema que se pretende resolver, existen muchas técnicas para entender una problemática específica, en consecuencia, las diversas disciplinas en el campo de las ciencias, han adoptado diferentes métodos para realizar un planteamiento acorde a sus necesidades.

En el caso puntual de esta tesis, se tomó como referencia un estándar de la industria para el desarrollo de software y, haciendo una sinergia con los conceptos tradicionales se presenta el siguiente modelo de investigación:

3.4.1. De campo y exploratoria.

A continuación se presenta un modelo generalizado y muy útil para el levantamiento de la información necesaria para el desarrollo de un sistema informático como solución a una problemática específica que consiste en desarrollar un esquema por etapas a partir del Modelo de Investigación de Congruencia Metodológica hasta llegar al punto donde se construyen las tablas y cuadros de congruencia metodológica que servirán de base a la etapa del análisis (determinación de requerimientos y especificaciones) y comprende:

- Identificación del problema
- Estudio previo del sistema
- Identificación de los factores que se involucran en el problema
- Factores controlables y no controlables
- Descripción del problema en relación a los factores
- Planteamiento relacional o funcional del problema
- Cuadro y tabla de congruencia metodológica
- Requerimientos y especificaciones para el desarrollo de software.

3.4.2. Identificación del problema

Para identificar la problemática, se realizaron entrevistas a los docentes: Ing. Janeth Franco, analista UPA, ing. Roberto Pico, director académico y Ab. Miriam Cárdenas, coordinadora de la carrera de sistemas, responsables de controlar la ejecución de las tareas y de reportar las actividades complementarias de los docentes. Los problemas más relevantes se describen a continuación:

Existen diversidad de formatos para la presentación y descripción de las actividades complementarias. Una de las principales razones es la falta de estandarización de documentos pues cada área elabora los formatos de acuerdo a su criterio y necesidades particulares.

El uso del papel es excesivo, debido a que tanto los docentes como los coordinadores de carrera imprimen y sacan fotocopias para sus archivos personales, entrega de informes, así como por la costumbre de mantener una constancia física de lo actuado.

Se estima que existe trabajo redundante en la compilación y tabulación de las horas cumplidas puesto que los docentes y coordinadores de carrera lo realizan por separado para su registro y control. Los informes son preparados en documentos Excel y posteriormente cotejados, situación que provoca demoras y uso de tiempos adicionales en actividades improductivas.

No existe una normalización en los procesos y mecanismos para el registro, entrega y almacenamiento de la información lo que, en ciertos casos, se torna complicado recabar y consolidar datos para la preparación de informes así como para las evaluaciones correspondientes.

3.5. Fuentes

3.5.1. Primarias

Las fuentes primarias, como es obvio, se seleccionaron a los funcionarios de la UTEQ con mayor relevancia en el proyecto, quienes conocen a profundidad la problemática y tienen claro las posibles soluciones.

En este caso, se destacan la ingeniera Janeth Franco, analista UPA, el ingeniero Roberto Pico, director académico de la UPA y la abogada Miriam Cárdenas, coordinadora de la carrera de sistemas, autoridades competentes y autorizadas, a quienes se las considera personal idóneo relativo a la investigación pues se evidencia que son protagonistas de primera mano para dar su testimonio de la problemática actual así como en orientar y recomendar las mejores soluciones al respecto.

3.5.2. Secundarias y electrónicas

Como fuentes secundarias se identificaron a trabajos similares sobre el tema, citas bibliográficas y recomendaciones las cuales se las describe en el capítulo 2 de este documento.

En lo referente a citas bibliográficas se enfatizó en encontrar las tecnologías más adecuadas para el desarrollo del software para aspirar que este proyecto se encuadre en los estándares de la industria en el desarrollo de software.

3.6. Población y muestra

3.6.1. Población

La población de esta investigación la conforman un total de 368 personas de la Universidad Técnica Estatal de Quevedo.

Cuadro 1. Población: Docentes y Autoridades de la UTEQ (2015)

Descripción	Cantidad
Decanos	6
Subdecanos	6
Vicerrector	1
Autoridades de la U.P.A.	2
Personal de Talento Humano	1
Coordinadores de Carrera	16
Docentes Titulares y Contratados	337
TOTAL	369

Fuente: Elaborado por el Autor

3.6.2. Muestra

La muestra con la que se trabajará correspondiente a coordinadores de carrera y docentes de la Universidad Técnica Estatal de Quevedo será del 19,51%.

Muestra de coordinadores de carrera y docentes.

Aplicamos la siguiente formula estadística para calcular la muestra de docentes.

$$n = \frac{N}{(E)^2 (N - 1) + 1}$$

En donde:

n = Tamaño de la Muestra

N = Población o Universo

E = Margen de error (0,1)

Cálculo de la muestra.

$$n = \frac{337}{(0.1)^2 (368) + 1}$$

$$n = \frac{337}{0.01 (368) + 1}$$

$$n = \frac{337}{4.68}$$

n = 72

La muestra a encuestar es de 72 personas de una población de 369 colaboradores en total.

3.7. Procedimiento Metodológico

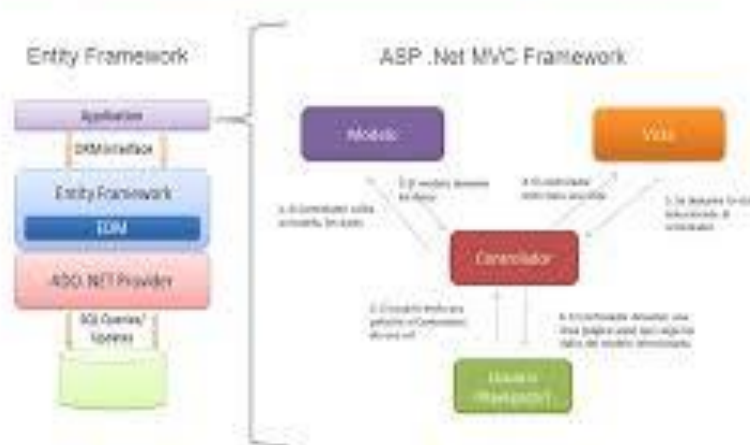
Para la construcción de este proyecto, se seleccionó la Metodología de desarrollo: Desarrollo Rápido de Aplicaciones RAD por sus siglas en inglés (Rapid Application Development), para utilizar el modelo de desarrollo CodeFirst que es un enfoque de la tecnología open source Entity Framework que plantea la creación de las clases POCO con un lenguaje de programación, en este caso C#, y definir las relaciones entre las mismas para que, después, la herramienta de diseño construya el modelo en la base de datos.

Lo importante es entender que con CodeFirst, lo primero es el código. En vez de comenzar creando la base de datos y después con ingeniería inversa generar las clases POCO (como lo hace Database First), con CodeFirst primero se crea el modelo con código y después se genera automáticamente la base de datos.

CodeFirst se considera la alternativa más adecuada para aquellos casos en los que hay que crear la base de datos desde cero en conjunto con la aplicación, y es el equipo de desarrollo quien debe hacerlo. Tal como es el caso de esta tesis, usando CodeFirst, se estima que no se debería escribir sentencias SQL (lo cual no implica que no es necesario conocer aspectos propios de su arquitectura). El enfoque es muy similar a lo que propone ModelFirst, pero en vez de trabajar con una herramienta de diseño se lo realiza directamente desde el código.

Ilustración 9. Asp.Net MVC y Entity Framework

ASP .Net MVC Framework y Entity Framework



Fuente: Microsoft, MSDN, 2011

3.7.1. Fase de determinación de requerimientos

Se determinan en esta fase los requerimientos que debe tener la aplicación web para analizar las necesidades operativas, administrativas y funcionales a nivel de autoridades, coordinadores de carrera y docentes, según las encuestas realizadas con cada uno de los representantes de los departamentos involucrados en este tipo de gestión de actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo.

3.7.2. Fase de análisis de la estructura del modelo de datos

Se define la estructura a crearse según los requerimientos de la información que se necesita almacenar electrónicamente, esta etapa da lugar también a la definición de las clases que se programarán según el modelo CodeFirst (Primero el código).

3.7.3. Fase de definición del tipo de arquitectura de desarrollo

En esta fase se escoge cuidadosamente el modelo de la arquitectura de desarrollo más idónea a fin de que la aplicación web sea mucho más fácil de entender y mantener esta aplicación n – capas, que permite definir cada una de ellas como un componente independiente y que puede ser reutilizado en otros proyectos de aplicaciones web, de la misma forma que permite aislar la capa de datos, para admitir el cambio del sistema de gestión de datos, con un marco de trabajo que está dirigido al uso de clases como entidades del modelo de datos (Entity Framework versión 6.1.1).

3.7.4. Fase de diseño

Se diseñaron las determinaciones de los requerimientos de la aplicación describiendo los tipos y modelos de componentes que se desarrollaron, tomando en consideración un conjunto de casos de uso, los cuales describen la forma de interactuar entre los usuarios y el sistema informático, en esta fase se elaboraron los documentos detallados en las especificaciones de requerimientos del sistema de esta tesis, que a continuación son:

- Introducción
- Propósito
- Alcance
- Definiciones, Acrónimos y Abreviaturas
- Referencias
- Funciones de la Aplicación
- Características de los Usuarios.

- Restricciones y limitaciones.
- Especificaciones funcionales (Casos de Uso o Contratos)

Subsiguientemente se diseñó la arquitectura y especificaciones técnicas, para determinar los componentes del sistema, y documentar en base a los diagrama de clases, diagrama entidad relación, diagramas de actividades y diagramas de secuencias.

3.7.5. Fase de Desarrollo

En el desarrollo de la aplicación se usó la herramienta Visual Studio 2013 con una interfaz de desarrollo de última tecnología que facilitó productivamente la creación del código y documentación del mismo, así como sus excelentes bondades en los procesos de ingeniería inversa en la obtención de diagramas. Cabe recalcar que el modelo de desarrollo CodeFirst (Primero el código) permite diseñar en el código la base de datos a implementarse la misma que es generada al momento de ejecutar la aplicación, lo cual permite tener un mejor control en cuanto a las modificaciones de la estructura de datos, porque estos se harían primero en el desarrollo de la aplicación y luego se replicarían en la base de datos mediante procesos de migración y actualización de la base de datos. De igual forma esta herramienta de desarrollo provee de algunos complementos que ayudan tanto en la parte estética como en la parte funcional de la aplicación, tal es el caso del complemento BootStrap versión 3.0 y los complementos de JQuery versión 1.10.2 que integra toda la potencia de Java Script utilizada en casi todas las aplicaciones web por su facilidad de escribir sentencias que se ejecutarán del lado del cliente, es decir al utilizar un explorador web en la ejecución de una aplicación interactiva.

3.7.6. Fase de Implementación

En la implementación de este prototipo se instaló una versión beta del software, se realizó la inducción a los usuarios, ejecutándose las debidas pruebas de funcionamiento y rendimiento. Esta fase es muy importante porque permite determinar la correcta funcionalidad del sistema, con usuarios como actores principales en el uso de un sistema informático, quienes dirán si realmente la

aplicación va a optimizar los procesos de control y gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo, y así realizar los respectivos ajustes a la aplicación.

CAPITULO IV
ANÁLISIS DE RESULTADOS

4.1. Análisis de Resultados

Mediante el sistema de encuestas realizado, definimos la amplitud que debería tener la aplicación web en el proceso de registro y control de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo, definiendo interrogantes puntuales como: ¿Está de acuerdo con el proceso actual de informe de actividades complementarias?, ¿Qué parte del proceso de registro y control de actividades complementarias considera el de mayor problema?, ¿Cree usted que el uso de una aplicación web optimizaría el informe de actividades complementarias?, ¿Está de acuerdo en utilizar una aplicación web para el registro de informes de actividades complementarias?, ¿Cree usted que usando una aplicación web los reportes de informes de actividades complementarias serían más rápido?, ¿Está usted de acuerdo en que el uso de una aplicación web reduciría el consumo de papelería?, ¿Usted cree que sería una ventaja utilizar una aplicación web disponible las 24 horas del día para el registro y control de los informes de actividades complementarias?.

No se plantean interrogantes como: ¿Usted conoce o ha usado el servicio de internet a través de un explorador?, ¿Tendría problemas usted para usar una aplicación web?, ¿Usted cuenta con el servicio de internet en su oficina o casa?, porque se conoce con anterioridad que el personal de docentes, coordinadores de carrera y autoridades de la Universidad Técnica Estatal de Quevedo, es un grupo totalmente idóneo con mucha experiencia en el uso de internet y de aplicaciones web y además el servicio de internet es uno de los servicios básicos utilizado por la institución.

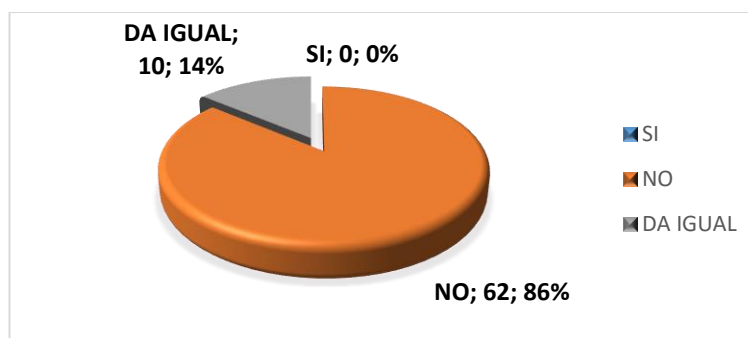
El grupo de docentes objetivo de la encuesta colaboró favorablemente al contestar las preguntas indicadas en un ambiente serio y formal, en un nivel alto de veracidad para obtener los siguientes resultados:

4.1.1. Informe de actividades complementarias.

Los datos que refleja el grafico 1, indica que el 86% de la población encuestada no está conforme con el proceso actual de registro y control de actividades

complementarias ya que conlleva la inversión de muchas horas de trabajo y recurso económico, y un 14% manifiestan estar de acuerdo con la forma de hacer los informes mensuales.

Gráfico 1. Satisfacción de sistema para informes de actividades complementarias

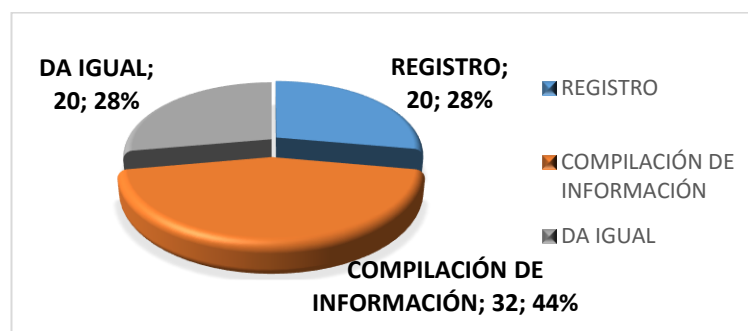


Fuente: UTEQ- UED
Elaborado por: El Autor

4.1.2. Problemas en el registro y control de actividades complementarias.

Los resultados indicados en el gráfico 2, muestra que el 44% de la población encuestada tienen problemas con el proceso de compilación de la información de los informes de actividades complementarias, debido a la cantidad de documentos que se deben procesar por cada docente y por cada una de las carreras de las facultades, mientras que un 28% indica dificultades en el proceso de registro de los informes por inconvenientes con los formatos de los archivos de registro de informes y el uso de sistemas no automatizados, y un 28% restante aprueban la forma actual del registro de informes.

Gráfico 2. Procesos que generan más problemas en el registro y control de actividades complementarias.

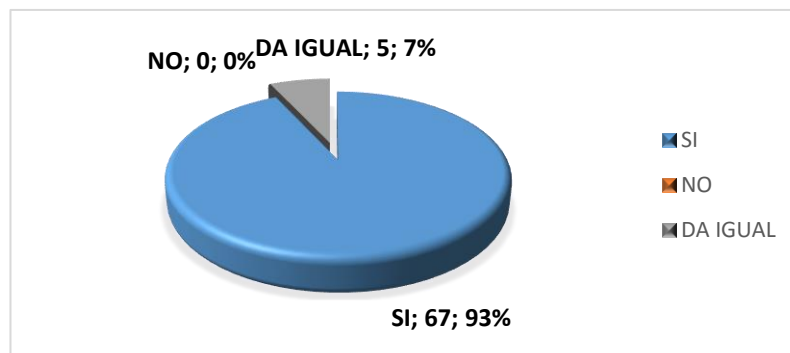


Fuente: UTEQ- UED
Elaborado por: El Autor

4.1.3. Aplicación web para el registro y control de informe de actividades complementarias.

Los valores de los resultados del gráfico 3, muestran que un 93% está de acuerdo en que los procesos de registro y control de actividades complementarias serían mucho más rápido utilizando un sistema informático con prestaciones que automatizarían las actividades de esta gestión, y un 7% está de acuerdo en realizar estas actividades ya sea con un sistema automatizado o procesos semi automatizados que es la forma actual de hacerlo.

Gráfico 3. Aplicación web para optimizar el registro y control de actividades complementarias.

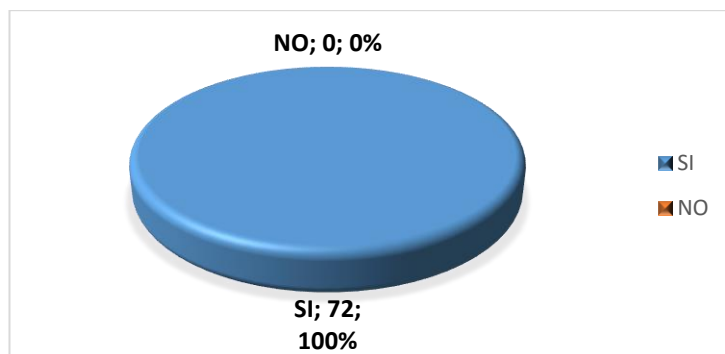


Fuente: UTEQ- UED
Elaborado por: El Autor

4.1.4. Aplicación web para el registro de informes de actividades complementarias.

Como indican los resultados del gráfico 4, el 100% de la población encuestada manifiesta una total aceptación en el uso de una aplicación web para optimizar los procesos de registro y control de los informes de actividades complementarias, ya que la mayoría ha manifestado experiencias agradables al utilizar este tipo de aplicaciones, debido a que los procesos se agilizarían enormemente.

Gráfico 4. Uso de aplicación web para el registro y control de actividades complementarias.

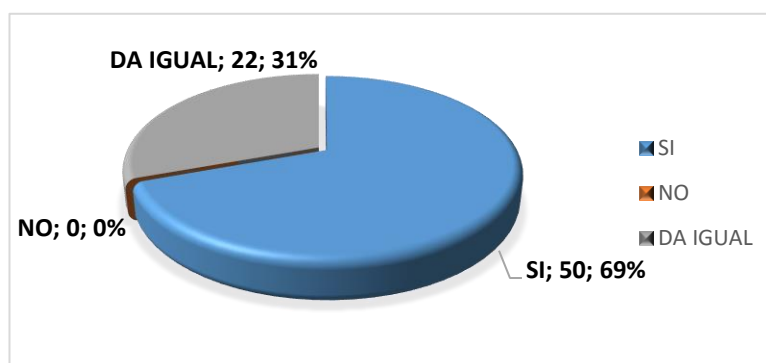


Fuente: UTEQ- UED
Elaborado por: El Autor

4.1.5. Emisión de reportes de informes de actividades complementarias.

Los resultados mostrados en el gráfico 5, indica que un 69% de la población encuestada confían en que un sistema informático es capaz de clasificar, ordenar, calcular todo tipo de información de una forma mucho más rápida que cualquier proceso manual, y la presentación de reportes de estos informes solo dependería del tiempo que se tomen los docentes en ingresar la información, y un 31% manifiesta estar de acuerdo con la forma actual de compilación y emisión de reportes de informes mensuales.

Gráfico 5. Emisión rápida de reportes de informes de actividades complementarias.



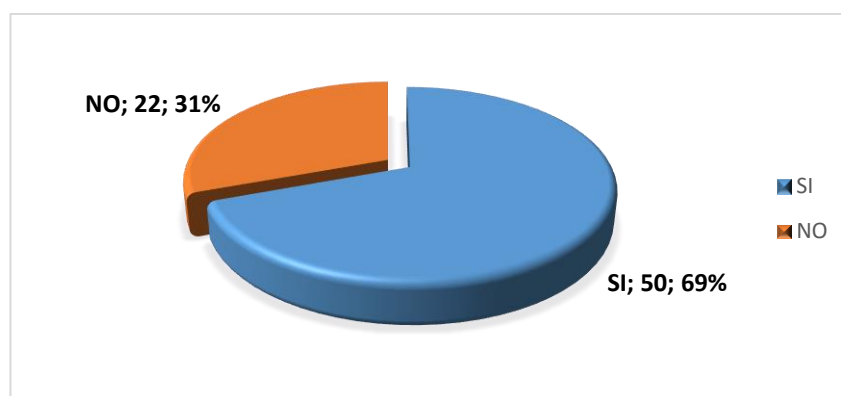
Fuente: UTEQ- UED

Elaborado por: El Autor

4.1.6. Reducción del consumo de papelería.

Los resultados del gráfico 6, indican que un 69% de la población encuestada está de acuerdo en que el uso de una aplicación web, reduciría en gran medida el consumo de papelería, debido a que confían plenamente en el registro electrónico de información, y un 31% manifiesta que prefiere tener un respaldo físico en papel como soporte de datos.

Gráfico 6. Reducción del consumo de papelería usando aplicación web.

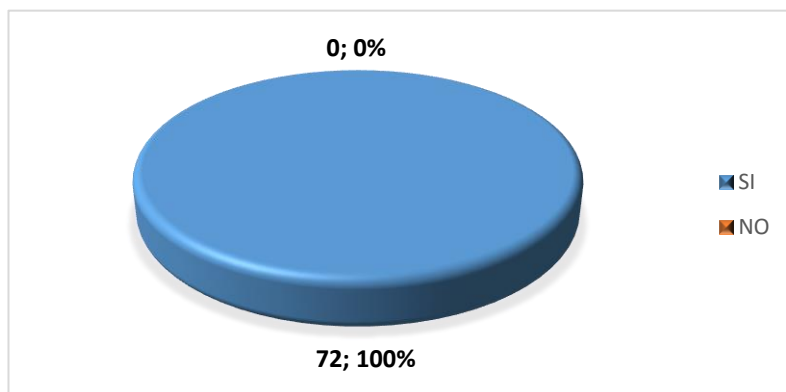


Fuente: UTEQ- UED
Elaborado por: El Autor

4.1.7. Ventajas de las aplicaciones web.

Los resultados mostrados en el gráfico 7, indican que el 100% de la población encuestada está de acuerdo en que sería muy ventajoso utilizar una aplicación web para el registro y control de actividades complementarias que garantice la disponibilidad a cualquier hora, manifestando que gracias a esta facilidad podrían aprovechar cualquier espacio de tiempo disponible para realizar este trabajo, para lo cual solo se necesitaría un computador y el servicio de internet, que en la actualidad son tecnologías accesibles y disponibles todo el tiempo,

Gráfico 7. Ventajas de la disponibilidad de las aplicaciones web



Fuente: UTEQ- UED
Elaborado por: El Autor

4.2. Discusión

Este estudio nos muestra resultados que evidencian la gran necesidad de automatizar los procesos de registro y control de los informes de las actividades complementarias realizadas por los docentes de la Universidad Técnica Estatal de Quevedo. Además, de acuerdo con los datos obtenidos en las fuentes primarias con los funcionarios de la Unidad de Planeamiento Académico (UPA), y con los coordinadores de carrera de la institución, se plantea por parte de ellos este requerimiento institucional manifestando el deseo de agilizar estos procesos mediante el desarrollo de una aplicación web, a fin de aprovechar las ventajas que brindan este tipo de aplicaciones por su disponibilidad y escalabilidad, esto se demuestra también en el gráfico No. 1 en el caso particular de este estudio, donde el 86% de los encuestados coinciden en estar inconforme con la forma actual de los procesos de registro y control de los informes de actividades complementarias, debido a problemas de gran amplitud, principalmente en los procesos de compilación de información y generación de reportes. REVILLA (2013) En su NewsLetter titulado “Las empresas sin sistemas automatizados tienen más problemas de rendimiento”, publica un estudio europeo sobre productividad empresarial llevado a cabo por Lexmark, donde se demuestra que las empresas que no usan sistemas automatizados tienen mayores problemas de rendimiento y menor productividad de los empleados y ciclos de trabajo más lentos, siendo uno de los mayores problemas los procesos administrativos manuales o semi automatizados por la propensión a los errores, de los 715 empleados de oficina de 12 países el 59% considera que los procesos manuales o semi automatizados son demasiados complicados al incluir varios pasos, un 70% considera que el tiempo dedicado a estos procesos se ha incrementado y que las empresas están desperdiciando un tiempo valioso.

También, en el nivel de aceptación de la implementación de una aplicación web como solución a los problemas de registro y control de actividades complementarias, el 93% de los encuestados manifiesta estar de acuerdo en que este tipo de software va a optimizar los procesos de registro y control de las actividades complementarias, confirmándose en el gráfico No. 4 donde el

100% de los encuestados aprueba el uso de una aplicación web en los procesos de registro y control de las actividades complementarias. MARTINEZ (2013) publica en su portal web “Ventajas de las aplicaciones web”, donde hace énfasis en las principales características de estas, indicando que son sencillas para los usuarios por la facilidad con que se puede acceder a estas, son sencillas y baratas de desarrollar por la infinidad de recursos baratos y gratuitos que actualmente existen en el medio, son sencillas y baratas de actualizar porque solo se actualizaría la aplicación en el equipo servidor, las necesidades de los computadores de los usuarios son menores porque la aplicación solo se ejecutaría en el equipo servidor, y por último las aplicaciones web facilitan la centralización de datos y su acceso desde cualquier parte del mundo.

De igual forma se prevé un ahorro considerable en el uso de papelería, como se indica en los resultados del gráfico No. 6, donde un 69% de las personas encuestadas, aprueban que al utilizar una herramienta informática segura, confiable que garantice la disponibilidad de la información en sistemas redundantes, eliminaría la necesidad de emitir documentos físicamente, porque estos serían analizados y aprobados electrónicamente e incluso podrían ser firmados digitalmente por los coordinadores de carrera a fin de garantizar mucho más la autenticidad de la información, esta automatización de los procesos de registro y control de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo, apoyadas con la implementación de tecnologías siendo en este caso a través de una aplicación web, redundaría en la creación de un sistema de información de tipo transaccional y gerencial que tiene como fin aumentar la eficiencia de estos procesos de información, asegurando una óptima gestión dentro de la institución, análisis que coincide con la definición técnica de LAUDON (2012), quien en su libro “Sistemas de Información Gerencial”, define al sistema de información como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones, coordinación y de control en una organización, ayudando a los directivos y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevas estrategias.

4.3. Análisis de situación actual

En esta etapa se desarrolla una narrativa para identificar los factores que se involucran o relacionan con la problemática presentada:

El artículo 117 de la Ley Orgánica de Educación Intercultural (LOEI) así como los reglamentos de la UTEQ como el Estatuto Orgánico de Gestión Organizacional por Procesos, especifican que, dentro de las 40 horas semanales, los docentes deben cumplir 20 horas pedagógicas y 20 horas de gestión en actividades complementarias, con el fin de cumplir la jornada laboral obligatoria. Estas actividades pueden ser: planificar actividades educativas, realizar reuniones con los demás docentes de su área, revisar tareas estudiantiles y evaluarlas, atender a los estudiantes, realizar actividades de recuperación o tutorías, asistir a cursos de nivelación, entre otras.

Para velar el cumplimiento de lo establecido, los docentes deben informar adecuadamente y de forma mensual, las actividades realizadas durante estos períodos de tiempo.

También es necesario controlar y evaluar su cumplimiento adecuadamente además, presentar informes periódicos a las autoridades competentes.

4.3.1. Identificación de los factores que se involucran en el problema

Por la necesidad de llevar un control adecuado del cumplimiento de esta norma, se crearon, espontáneamente y sin una estandarización previa, procedimientos e informes de acuerdo a las necesidades de cada facultad. Como consecuencia de esto, resulta muy complicado recopilar la información con la calidad y oportunidad requerida. Un ejemplo es que los docentes, para reportar sus actividades complementarias, deben presentar a los coordinadores de carrera informes impresos con el detalle de lo actuado durante cada mes. Los coordinadores de carrera, por su parte, deben entregar una certificación de lo recibido a cada docente que consiste en una fotocopia del informe con su firma de recepción, compilar y tabular la información recibida con el fin de, posteriormente, emitir informes para sus decanos y, éstos a su vez, a los

directores académicos de la Unidad de Planeamiento Académico UPA, quienes, por su parte, previa revisión y análisis, los transfieran al departamento de Talento Humano para sus cálculos salariales y evaluaciones de desempeño.

De lo expuesto, para el desarrollo del proyecto, se estima que se incluyan y se den solución a los siguientes problemas:

1. Estandarización de la información requerida.
2. Recopilación ágil de las actividades complementarias de los docentes
3. Facilidad y comodidad para el registro
4. Minimizar esfuerzos y tiempos de registro de la información
5. Oportunidad en la entrega de los informes a las autoridades competentes

4.3.2. Factores controlables y no controlables

De todos los factores descritos, se especifican qué factores son controlables y cuáles no:

4.3.2.1. Factores Controlables:

1. Estandarización de la información requerida:

Al diseñar formularios estándar y únicos para todas las facultades, se pretende que la información requerida sea común, debido a que las actividades complementarias son similares así como sus objetivos finales, para evitar de ese modo la complejidad de formularios específicos que, eventualmente, pueden distorsionar el alcance de esta metodología.

2. Recopilación ágil de las actividades complementarias de los docentes:

Se espera que la información a recopilar sea común, estándar y la mínima requerida con el fin de facilitar el registro, para evitar en lo posible información complementaria que complique su recopilación.

Como resultado de la estandarización se estima que se facilite el proceso de aprendizaje y adaptabilidad de los docentes y autoridades al sistema para que, en muy pocos pasos fáciles de recordar y ejecutar, se consigan los objetivos institucionales en este requerimiento.

3. Facilidad y comodidad para el registro

El sistema será lo suficientemente intuitivo y sencillo, con pocos pasos de navegación y desde cualquier sitio que disponga de una computadora y conexión a internet, los docentes, en cualquier momento antes de las fechas máximas establecidas, puedan realizar el registro de sus actividades. Es más, se sugiere motivar a que los docentes cumplan con el registro mientras realizan las actividades complementarias, con lo cual se conseguirá información actualizada en línea y en el momento justo.

4. Oportunidad en la entrega de los informes a las autoridades competentes

Como consecuencia de lo anterior, las autoridades tendrán su información inmediatamente y en línea, pues si los docentes ingresan adecuadamente sus informes mientras estos se registren, el sistema procesará los datos requeridos, prácticamente en forma simultánea.

5. Minimizar esfuerzos y tiempos de registro de la información:

Con la implementación de este proyecto, el trabajo de los coordinadores de carrera y demás autoridades se limitan al nivel de supervisión y control, eliminando los trabajos repetitivos de digitación, compilación, tabulación y elaboración de informes.

4.3.2.2. Factores no controlables:

Existen factores que fueron mencionados en este relevamiento que están fuera del alcance del proyecto, pues dependen de la responsabilidad personal de los involucrados, así como del seguimiento y control de las autoridades competentes que, entre otros, son:

1. Fidelidad de la información entregada:

Se estima que los docentes tienen el profesionalismo para reportar exclusivamente lo que realizaron tanto en tiempo como en veracidad de lo actuado.

2. Revisión y análisis de los decanos y directores académicos de la Unidad de Planeamiento Académico UPA

Es responsabilidad de las mencionadas autoridades, periódicamente revisar y analizar los informes elaborados por el sistema con el fin de convalidar el proceso así como certificar su calidad y exactitud.

3. Cálculos salariales y evaluaciones de desempeño por parte de Talento Humano.

El sistema se limita a entregar la información requerida por Talento Humano con relación al registro de las actividades complementarias de los docentes. El uso y fines de dicha información escapan de los objetivos del sistema y será de responsabilidad exclusiva de Talento Humano y las autoridades correspondientes.

4.3.3. Descripción del problema en relación a los factores

La hipótesis planteada para la ejecución de la tesis es la siguiente:

“La aplicación web optimizará los procesos de registros, control y gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo”.

Como se ha dicho en algunas ocasiones, la aplicación web es sólo una variable para la consecución de las expectativas creadas con la hipótesis. El resto de variables analizadas tales como: Comportamiento, Capacitación, Seguimiento, Control y Retroalimentación son elementos indispensables para convertir la hipótesis en una idea verdadera o propósito cumplido y responder de forma eficaz a la problemática planteada con bases científicas.

4.3.4. Planteamiento relacional o funcional del problema

De la etapa anterior, a continuación, se describen cada una de las variables involucradas para el éxito del proyecto:

El comportamiento de los docentes en cumplir con lo establecido de una manera verás, oportuna y proactiva es la base fundamental para el éxito del proyecto pero, para esto, es necesario que reciban una capacitación adecuada,

no sólo en el uso del software sino en todo el proceso para su fiel cumplimiento.

El seguimiento y control por parte de los decanos y directores académicos UPA también son necesarios, pues, como en todo proceso, es indispensable la supervisión continua así como tomar las medidas correctivas oportunamente con el fin de mantener un flujo del proceso tal como se espera.

Finalmente, la retroalimentación es vital para el mantenimiento productivo de un proyecto en el tiempo. Todos los involucrados en el proyecto, deben evaluar continuamente todas las actividades intervinientes y proponer cambios y mejoras cuando lo ameriten. También es responsabilidad de las autoridades competentes escuchar estas propuestas y ponerlas en práctica si lo estiman. Requerimientos y especificaciones para el desarrollo de software.

Como objetivo principal de esta tesis, se planteó y aprobó, el desarrollo de un software que sirva como herramienta operativa para el registro y control de la gestión de las actividades complementarias de los docentes de la UTEQ.

Los requerimientos principales son:

- Que el sistema permita el registro de los usuarios del sistema, a los cuales se les designe un perfil específico de acuerdo a las funciones que tendrán dentro del sistema.
- Que el sistema permita a los docentes registrar sus actividades complementarias en un formato estándar, ágil y sin complicaciones.
- Que los decanos y directores académicos UPA puedan acceder al sistema y emitir reportes para seguimiento y control de lo actuado por los docentes.
- Que Talento Humano disponga de la información requerida de las actividades complementarias de los docentes para sus variables salariales y evaluación de desempeño correspondientes.

4.3.5. Bibliográfica

Actualmente la evolución acelerada de las Tecnologías de la Información genera cada vez nuevas y mejores herramientas, arquitecturas y metodologías para el desarrollo de software, lo que requiere una constante investigación de

las tendencias de la industria en este aspecto. Un procedimiento para mantenerse actualizado es con la investigación bibliográfica y documental.

En el trabajo de investigación bibliográfica se buscaron los conceptos y tecnologías más novedosos respecto a la forma de encarar un desarrollo de este software, como por ejemplo la utilización del patrón de diseño modelo vista controlador (MVC) versión 5 (ver cita bibliográfica), herramientas poderosas para la planificación, modelado, desarrollo y testeo de aplicaciones como el Visual Studio 2013, otra herramienta muy utilizada actualmente encargada de la parte estética de las aplicaciones web y que se integra automáticamente en los proyectos de desarrollo de Visual Studio es Bootstrap versión 3.0. De igual forma, el Visual Studio ya integra como parte de un proyecto de desarrollo de aplicaciones ASP.NET MVC los plugin de JQuery conjuntamente con todo el poder de las herramientas Ajax, utilizados en las validaciones de datos del lado del cliente y el tratamiento de información de una forma asincrónica.

En base a los horarios de actividades complementarias con que deben cumplir los docentes de la Universidad Técnica Estatal de Quevedo, una vez realizadas estas actividades, deben ser informadas por medio de reportes hecho en el programa Microsoft Excel para luego ser impreso y fotocopiado con el fin de entregarlos a sus respectivos coordinadores de carrera, quien recibirá y certificará la recepción del mismo con su firma en la copia del informe. El coordinador de carrera deberá compilar la información por cada uno de los docentes y por cada tipo de actividad, para luego crear y emitir un reporte resumido hecho en el mismo programa (Microsoft Excel) y entregarlo a las autoridades correspondiente: Facultades y Directivos de la UPA de la institución para su respectivo análisis y control, tareas que al ser automatizadas mejorarían enormemente la calidad de control y gestión.

4.3.6. Análisis del proceso anterior de registro y control de Actividades Complementarias.

De acuerdo con los datos especificados tanto por los directivos, coordinadores de carrera y docentes de la Universidad Técnica Estatal de Quevedo, tenemos

el siguiente cuadro que indica el tiempo promedio estimado en cada uno de los procesos en el registro y control de actividades complementarias.

Cuadro 2. Promedio mensual del tiempo usado con el proceso anterior

TIEMPOS DE TRABAJO	
PROCESO	PROMEDIO EN HORAS
Registro de Informe de Actividades (Docentes).	3 a 4
Recepción y Revisión de Informes de Actividades (Coordinadores de Carrera).	1
Compilación de Información de Informes de Actividades (Directivos Facultades).	15 a 20
Entrega de Informes de los Docentes a los Coordinadores de Carrera.	Depende de la presencia del Coordinador de Carrera.*

Fuente: UTEQ- UED

Elaborado por: El Autor

* El tiempo promedio estimado en el proceso de entrega de informes de actividades de los docentes a los coordinadores de carrera se ve afectado por la no presencia del coordinador en su oficina, quien tiene que asistir a reuniones de trabajo, comisiones, entre otras actividades que ocasionan su ausencia del puesto habitual.

Bajo este mismo esquema de trabajo en el proceso de registro y control de actividades complementarias, podemos decir que existe un buen consumo de algunos recursos materiales por el tipo de control aplicado, indicado en el siguiente cuadro.

Cuadro 3. Costo del consumo mensual de recursos materiales con el proceso anterior

COSTOS

MATERIAL	CANTIDAD/CONSUMO	COSTO UNIT.	COSTO TOTAL
PAPEL A4	3253*	0.01	32.53
TINTA DE IMPRESORA	1 REPOSICIONES EN SISTEMAS DE ALIMENTACIÓN CONTINUA.	9.00	9.00
COMPUTADORA	1*	0.00	0.00
COSTO TOTAL USD.			41.53

Fuente: UTEQ- UED

Elaborado por: El Autor

* Se estima que los 357 docentes tienen tres (3) actividades complementarias, por lo tanto cada uno debe consumir un mínimo de nueve (9) hojas de papel tamaño A4, se considera que por cada actividad debe emitir un informe mensual y por cada informe debe tener un duplicado para asegurar la constancia de la recepción del documento. De igual forma los directivos de cada una de las facultades deben compilar la información y emitir los respectivos informes que serán entregados a la unidad de Talento Humano, utilizando para este efecto cinco (5) o más hojas de papel tamaño A4.

* Las personas que intervienen en estos procesos de registro y control de actividades complementarias, cuentan con su propia computadora, además la institución provee estos equipos en sus respectivos puestos de trabajo.

* 4000 Páginas es el rendimiento total basado en una botella de tinta negra en impresoras Epson con sistemas de tinta continua. Rendimiento basado en el patrón ISO/IEC 24712, con metodología EPSON. Tomado del sitio web: <http://global.latin.epson.com/Catalog/Epson-L350/Resumen>.

En el siguiente cuadro se indican los requerimientos principales en el registro y control de actividades complementarias, por parte de los docentes y coordinadores de carrera, así como ciertos inconvenientes durante el proceso de entrega y revisión de la información.

Cuadro 4. Requisitos e inconvenientes en el registro y control usando el proceso anterior

REQUISITOS E INCONVENIENTES EN EL REGISTRO Y CONTROL DE ACTIVIDADES COMPLEMENTARIAS	
PROCESO	REQUISITO/INCONVENIENTE
Registro de Informe de Actividades Complementarias (Docentes).	<ul style="list-style-type: none"> • Requiere de un software adicional (Excel). • Tiene que crear el formato de acuerdo al tipo de actividad a informar. • Tiene que crear un nuevo archivo por cada mes y tipo de actividad a informar.
Entrega de Informe de Actividades Complementarias (Docentes).	<ul style="list-style-type: none"> • Depende de la disponibilidad del coordinador de carrera.
Seguridad de los datos.	<ul style="list-style-type: none"> • Documentos extraviados (tiene que volver a imprimir) • Archivos electrónicos en soporte de datos móviles que se pueden perder fácilmente.
Revisión y compilación de Informe de Actividades Complementarias (Directivos Facultades).	<ul style="list-style-type: none"> • Documentos del mismo tipo de actividad tienen algunas diferencias en el formato.

Fuente: UTEQ- UED

Elaborado por: El Autor

4.3.7. Análisis del proceso de registro y control de Actividades Complementarias después del desarrollo y ejecución de la aplicación web.

Así como se indicó anteriormente los promedios de tiempo estimado, los recursos y materiales utilizados, y los inconvenientes durante los procesos de registro y control de actividades complementarias, de igual forma a continuación se detallan los resultados después del desarrollo y ejecución de la

aplicación web, a fin de poder establecer las comparaciones entre las dos formas de trabajo.

Cuadro 5. Promedio mensual de tiempo usando la aplicación web para el registro y control de actividades complementarias

TIEMPOS DE TRABAJO	
PROCESO	PROMEDIO EN HORAS
Registro de Informe de Actividades (Docentes).	2
Recepción y Revisión de Informes de Actividades (Docentes y Coordinadores de Carrera)	1
Compilación de Información de Informes de Actividades (Directivos Facultades).	0
Análisis de la información presentada en los informes de actividades (Directivos Facultades).	1 *
Entrega de Informes de los Docentes a los Coordinadores de Carrera.	0

Fuente: UTEQ- UED

Elaborado por: El Autor

* El uso de un sistema informático requiere en ciertos casos que la información procesada sea analizada y avalorada, más aún cuando de esta dependen los cálculos de haberes y proviene de otros departamentos de la institución.

Cuadro 6. Costo del consumo mensual de recursos materiales usando la aplicación web para el registro y control de actividades complementarias

COSTOS			
MATERIAL	CANTIDAD/CONSUMO	COSTO UNIT.	COSTO TOTAL

PAPEL A4	200*	0.01	2.00
TINTA DE IMPRESORA	0.1 REPOSICIONES EN SISTEMAS DE ALIMENTACIÓN CONTINUA.	9.00	0.90
COMPUTADORA	1*	0.00	0.00
COSTO TOTAL USD.			2.90

Fuente: UTEQ- UED

Elaborado por: El Autor

* Se estima un consumo mínimo de 200 hojas de papel tamaño A4, debido a los informes que emitirían los coordinadores de carrera, directivos de las facultades y talento humano, así como ciertos reportes que necesitarían imprimir los docentes sobre el resumen de sus horas cumplidas en las actividades complementarias.

Cuadro 7. Requisitos e inconvenientes en el registro y control usando la aplicación web para el registro y control de actividades complementarias

REQUISITOS E INCONVENIENTES EN EL REGISTRO Y CONTROL DE ACTIVIDADES COMPLEMENTARIAS	
PROCESO	REQUISITOS/INCONVENIENTE
Registro de Informe de Actividades Complementarias (Docentes).	<ul style="list-style-type: none"> • No requiere el uso de Excel. • No tiene que crear ningún formato para ningún tipo de actividad a informar. • No tiene que crear ningún archivo nuevo.
Entrega de Informe de Actividades Complementarias (Docentes).	<ul style="list-style-type: none"> • No tiene que entregar ningún documento.
Seguridad de los datos.	<ul style="list-style-type: none"> • No es necesario imprimir los informes de actividades por parte de los docentes. • No es necesario tener la información en soporte de

	datos móviles, puesto que están almacenados en una base de datos segura.
Revisión y compilación de Informe de Actividades Complementarias (Directivos Facultades).	<ul style="list-style-type: none"> No hay diferencia en los formatos de ingreso de datos de las actividades complementarias.

Fuente: UTEQ- UED

Elaborado por: El Autor

4.3.8. Cuadro Comparativo

Cuadro 8. Comparativo entre el proceso anterior y después del desarrollo y ejecución de la aplicación web.

Dimensión	Indicador	Observación aplicando el proceso anterior	Observación después del desarrollo y ejecución de la aplicación web.
EFICIENCIA	Tiempo en el registro de informe de actividades.	4 h.	2 h.
	Tiempo en la recepción y revisión de informes de actividades.	1 h.	1 h.
	Tiempo en la compilación de información de informes de actividades.	15 a 20 h.	0 h.
	Tiempo en el	0 h.	1 h.

	análisis de la información presentada en los informes de actividades.		
	Tiempo en la entrega de los informes de los docentes a los coordinadores de carrera.	Tiempo no estimado, depende de la disponibilidad del coordinador de carrera.	0 h.
ECONOMIA	Costo del proceso mensual.	USD. 41.53	USD. 2.90
IMPACTO AMBIENTAL	Consumo de agua limpia. En fabricación de papel.	1203.61 lt.	74 lt.
SEGURIDAD	Integridad de los Datos.	Datos propensos a perderse al estar en soportes móviles. Pueden sobrescribirse accidentalmente.	Datos totalmente seguros en un servidor central garantizando la disponibilidad.
SATISFACCIÓN	Registro de Actividades Complementarias (Docentes).	Incomodo, requiere el uso de software adicional (Excel). Debe crear formatos y nuevos archivos cuidadosamente.	Proceso ágil, solo necesita un programa explorador e internet. No necesita crear formatos, la aplicación los

		Tiene que imprimir obligatoriamente el documento por duplicado.	provee. No necesita imprimir los documentos (información compartida).
	Revisión y Compilación de Información de Informes.	Demora mucho tiempo, alrededor de 15 a 20 horas de trabajo.	Demasiado ágil, solo se necesita 1 hora para la revisión y emisión de reportes.

Fuente: UTEQ- UED

Elaborado por: El Autor

4.4. Especificación de Requerimientos

4.4.1. Introducción

Como parte integrante y medular de la tesis de grado, se propuso y aprobó el desarrollo de una aplicación web para el control de la gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo

4.4.2. Propósito

El propósito de la aplicación web para el control de la gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo es poner a disposición de la docencia una herramienta que proporcione formularios estándar y únicos para todas las facultades, fáciles de entender e interpretar para evitar, en lo posible, la complejidad de formularios específicos que, eventualmente, pueden distorsionar el objetivo del proyecto desmotivando su uso y aplicación.

Como resultado de la estandarización se estima que se facilite el proceso de aprendizaje y adaptabilidad de los docentes y autoridades al sistema para que,

en muy pocos pasos fáciles de recordar y ejecutar, se consigan los objetivos institucionales en este requerimiento.

El sistema será lo suficientemente intuitivo y sencillo para que, con muy pocos pasos de navegación y desde cualquier sitio que disponga de una computadora y conexión a internet, los docentes, en cualquier momento antes de las fechas máximas establecidas, puedan realizar el registro de sus actividades. Es más, se sugiere motivar a que los docentes cumplan con el registro mientras realizan las actividades complementarias, con lo cual se conseguirá información actualizada en línea y en el momento justo.

Como consecuencia de lo anterior, las autoridades tendrán su información inmediatamente y en línea, pues si los docentes ingresan adecuadamente sus informes mientras los están registrando, el sistema procesará los datos requeridos, prácticamente en forma simultánea.

Con la implementación de este proyecto, el trabajo de los coordinadores de carrera y demás autoridades se limitan al nivel de supervisión y control, eliminando los trabajos repetitivos de digitación, compilación, tabulación y elaboración de informes

4.4.3. Alcance

El alcance del proyecto es el diseño de la aplicación web para el registro de las actividades complementarias de los docentes con los siguientes módulos: (La descripción detallada de cada módulo, se describen en 4.4.3. Funciones del Producto).

Iniciar sesión:

Registro de Datos Iniciales

- Tipos de identificación:
- Tipos de Relación laboral:
- Tipos de Dedicación de tiempo:
- Roles:
- Usuarios:

- Asignación de Roles:
- Tipos de Actividades:
- Asignación de Actividades:

Facultades y Carreras

- Datos de Facultades:
- Datos de Carreras:

Períodos Lectivos

- Períodos:

Horarios de Docentes

- Aprobar Horario:

Informe de Actividades

- Aprobar Informe:
- Reporte de Horas:

Horario de Actividades

- Mantener Horario de Actividades:

Informe de Actividades

- Crear Informes
- Consulta y Edición de Informes:
- Reporte de Horas Cumplidas
 - Informe a las autoridades competentes:

4.4.4. Definiciones, Acrónimos y Abreviaturas

UPA:

Unidad de Planeamiento Académico.

W3C:

World Wide Web Consortium, Consorcio del ancho mundo de la red (internet).

HTTP:

Hiper Text Transfer Protocol, Protocolo de transferencia de hiper texto.

DNS:

Domain name System, Sistema de nombres de dominio.

CSS:

Cascade Style Sheet, Hojas de estilos en cascada utilizados en el diseño de páginas web.

JS:

Java Script File, Archivo script escrito en lenguaje de programación java.

EDMX:

Entity Data Model Xml, modelo de entidades de datos basado en xml.

LINQ:

Language Integrated Query, Lenguaje integrado de consultas.

API:

Application Programming Interface, Interfaz de programación de Aplicaciones.

PANTALLA MODAL:

Interfaz gráfica con preferencia de vista y acción sobre el resto de pantallas.

REFACTORIZAR:

Proceso que consiste en mejorar un código escrito, cambiando su estructura interna sin modificar su comportamiento externo.

4.4.5. Referencias

Anteproyecto de tesis de grado.

ANSI/IEEE Std. 830-2007 actualizado Guía del IEEE para la Especificación de Requerimientos de Software.

Resúmenes de reuniones de trabajo con las autoridades descritas en “Fuentes primarias”.

Investigación de Campo y exploratoria.

4.5. Descripción General

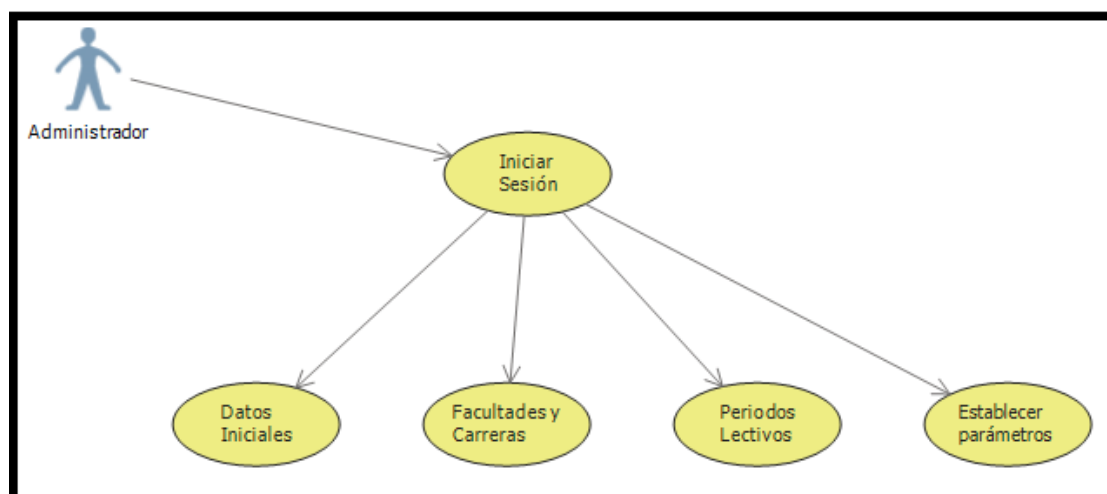
4.5.1. Perspectiva del producto

Con la implementación de la aplicación se aspira que sea una herramienta útil y eficaz para el registro de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo. Además que reduzca significativamente las actividades de registro y control realizadas actualmente, pues, con el oportuno ingreso de información al sistema, se estima que se sustituirán totalmente. Adicionalmente, se espera que los informes de control también se reemplacen por los proporcionados por esta aplicación para aportar de esta manera, con información relevante en el control y evaluación de desempeño correspondiente en estas actividades.

4.5.2. Funciones del producto

Sobre la base del alcance del proyecto, se describen las funciones de la aplicación:

Ilustración 10. Funciones del Administrador



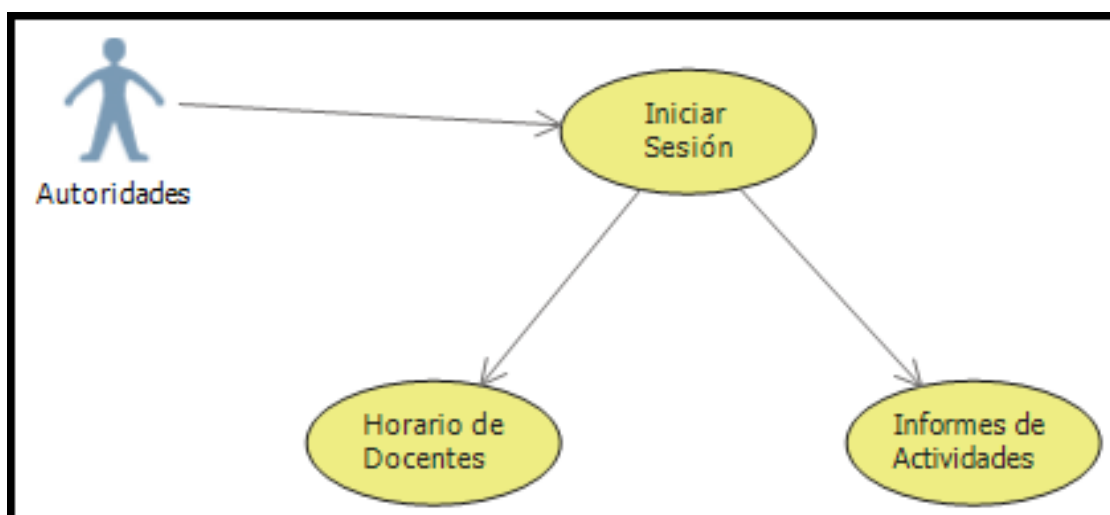
Fuente: Elaborado por el Autor

Ilustración 11. Funciones de los Docentes



Fuente: Elaborado por el Autor

Ilustración 12. Funciones de las Autoridades



Fuente: Elaborado por el Autor

4.5.2.1. Iniciar sesión

Este módulo es un filtro de seguridad para acceso al sistema, solicita al usuario el ingreso de su identificación y clave, convalida lo digitado y si coincide con la información registrada en el sistema arma el perfil correspondiente para la navegación en el sistema.

4.5.2.2. Datos Iniciales

Aquí se registra información general del sistema y que servirán para armar listas de opciones que el usuario podrá seleccionar durante el registro de sus

actividades y que son necesarias para evitar errores de conceptualización o digitación.

- **Tipos de identificación**

Se registran las posibles identificaciones que puedan tener los docentes ej.: cédula de ciudadanía, pasaporte o RUC.

- **Tipos de Relación laboral**

Se listan los posibles contratos de trabajo que tienen los docentes con la UTEQ.

- **Tipos de Dedicación de tiempo**

Se especifican las horas que los docentes deben dedicar a las actividades complementarias.

- **Roles**

Se definen las funciones que se pueden realizar en el sistema.

- **Usuarios**

Para registrar a las personas, docentes y autoridades, que pueden acceder al sistema.

- **Asignación de Roles**

Se definen los roles que van a tener cada uno de los usuarios en el sistema y que fueron creados en la opción de **Roles**.

- **Tipos de Actividades**

Se especifican que las actividades complementarias que realizarán los docentes.

- **Asignación de Actividades**

Se definen las actividades complementarias para cada uno de los docentes.

4.5.2.3. Facultades y Carreras

En el levantamiento de requerimientos, (ver fuentes primarias) se estableció la necesidad de discriminar a la UTEQ en varios tipos de organizaciones de acuerdo a sus objetivos académicos y se la dividió en:

- **Datos de Facultades**

Se registran a las facultades e información complementaria.

- **Datos de Carreras**

Igual que facultades, se registran las diferentes carreras e información básica.

4.5.2.4. Períodos Lectivos

- **Períodos**

Identifican los períodos lectivos en que se aplican los registros de las actividades complementarias.

4.5.2.5. Horarios de Docentes

- **Aprobar Horario**

Se deben registrar los horarios aprobados por los coordinadores de carrera.

4.5.2.6. Informe de Actividades

- **Aprobar Informes**

Se deben registrar los informes aprobados por los coordinadores de carrera.

- **Reporte de Horas:**

Consulta de horas aplicadas en actividades registradas por los docentes.

4.5.2.7. Horario de Actividades

- **Crear Informes**

Se deben registrar los informes de actividades realizadas por los docentes.

- **Consulta y Edición de Informes**

Permite la consulta y edición de actividades realizadas por los docentes.

- **Reporte de Horas Cumplidas**

Consulta de horas cumplidas en actividades registradas por los docentes.

4.5.3. Características de los Usuarios

Se definen tres tipos de usuarios:

- Administradores del sistema
- Docentes
- Autoridades

4.5.3.1. Administradores del sistema:

Son usuarios expertos encargados de configurar el sistema, quienes, a más de instruir a los docentes y autoridades en la funcionalidad del sistema, son los encargados de parametrizarlo.

4.5.3.2. Docentes

Son los usuarios quienes crean sus horarios de actividades y registran sus actividades complementarias.

4.5.3.3. Autoridades

Usuarios que tienen acceso únicamente de consulta para emitir sus reportes y en el caso de que la autoridad sea un coordinador de carrera, tendrá la opción de aprobar los horarios y los informes de actividades registrados por los docentes.

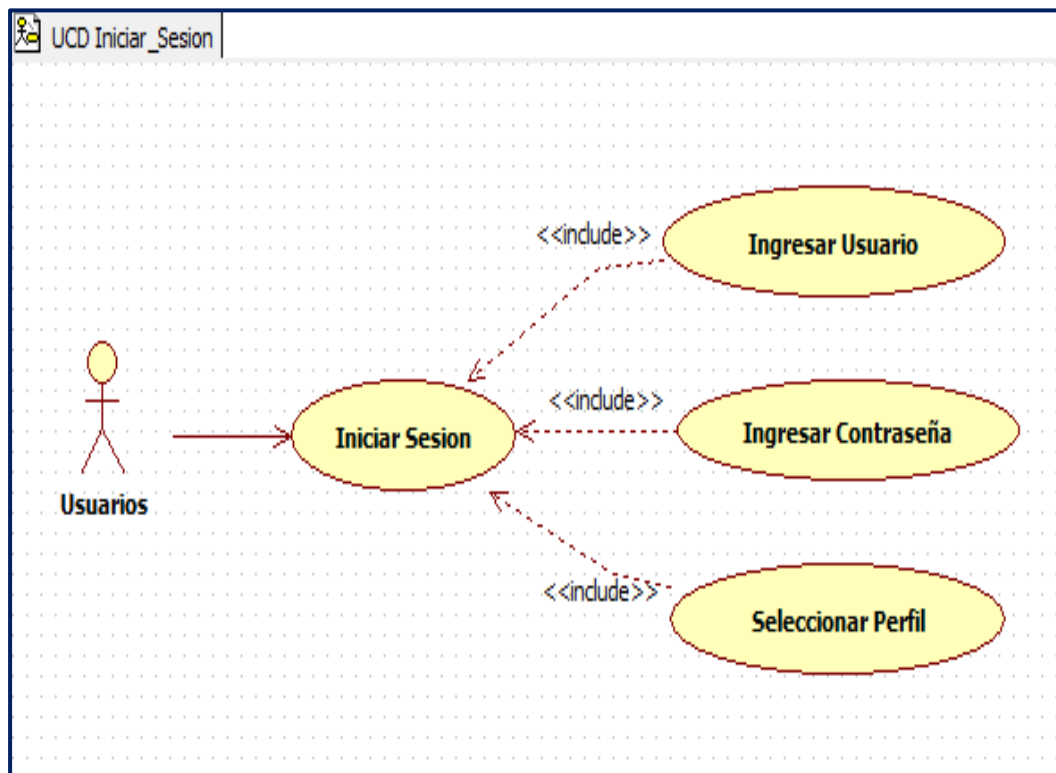
4.6. Requerimientos Específicos

4.6.1. Requerimientos Funcionales

4.6.1.1. Casos de Uso

Los Diagramas de Casos de Usos, utilizados en la etapa de estudio del proyecto de desarrollo de un sistema informático, dan la pauta para entender la funcionalidad e interactividad del sistema con el usuario, definiéndose las particularidades que la aplicación debe realizar durante su ciclo de vida en el modo de producción. Para cumplir con este requisito de ingeniería de software se usó la herramienta de modelados open source Start Uml/Mda versión 5.0.2.

Ilustración 13. Caso de Uso - Iniciar sesión



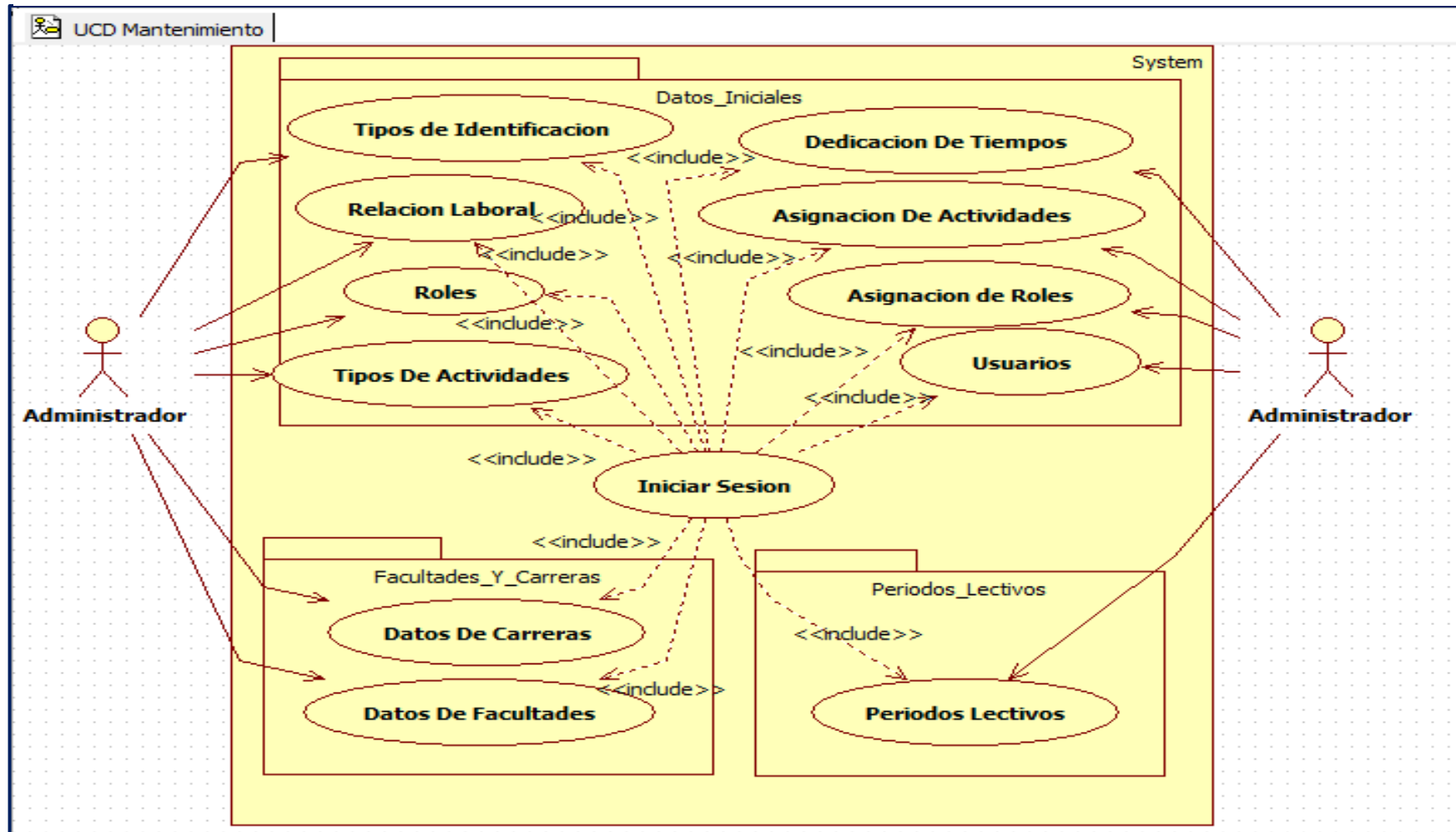
Fuente: Elaborado por el autor

Cuadro 9. Caso de uso - Iniciar Sesión

Código	UC-001	
Nombre del caso de uso	Iniciar Sesión	
Descripción	Inicio de Sesión	
Actores	Administrador, Autoridades, Docentes	
Pre-Condiciones	El usuario debe estar registrado en la base de datos como Administrador, Autoridad o Docente.	
Flujo Principal	1. El usuario da clic en la opción Iniciar Sesión de la aplicación.	2. El sistema muestra la pantalla de ingreso de usuario y clave.
	3. El usuario ingresa sus datos de identificación y clave de acceso y da clic en Aceptar.	4. El sistema verifica las credenciales y si el usuario tiene más de un perfil presenta las opciones de selección del rol a usar.
	5. El usuario elige el rol y da clic en aceptar	6. El sistema presenta el menú correspondiente al rol seleccionado.
Flujo Alternativo	El sistema valida la información, y en caso de ser incorrecta presenta un mensaje de error. El usuario presiona clic en Cancelar, el sistema cierra la pantalla y no se realiza ninguna acción.	
Pos-Condiciones	Las credenciales del usuario han sido aceptadas en el sistema.	

Fuente: Elaborado por el Autor

Ilustración 14. Casos de Uso - Mantenimiento Datos Iniciales



Fuente: Elaborado por el Autor

Cuadro 10. Caso de uso - Crear Tipos de Identificación

Código	UC-002	
Nombre del caso de uso	Crear Tipos de Identificación.	
Descripción	Creación de Tipos de Identificación.	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario selecciona la opción Tipos de Identificación del menú de Datos Iniciales.	2. El sistema muestra las opciones de la opción Tipos de Identificación y una lista de los registros existentes.
	3. El usuario selecciona la opción Crear Nuevo Registro.	4. El sistema muestra la página de edición de registro para el ingreso de la información.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Tipos de Identificación.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página actual sin realizar ninguna acción.	
Pos-Condiciones	El Tipo de Identificación ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 11. Caso de uso - Crear Relación Laboral

Código	UC-003	
Nombre del caso de uso	Crear Relación Laboral	
Descripción	Creación de Relación Laboral	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario selecciona la opción Tipos de Relación Laboral del menú de Datos Iniciales.	2. El sistema muestra las opciones de Tipos de Relación Laboral y una lista de los registros existentes.
	3. El usuario hace clic en la opción Crear Nuevo Registro.	4. El sistema muestra la página de registro para el ingreso de los datos del nuevo registro.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Tipos de Relación laboral
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página actual sin realizar ninguna acción.	
Pos-Condiciones	El Tipo de Relación Laboral ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 12. Caso de uso - Crear Tipo de Dedicación de Tiempo

Código	UC-004	
Nombre del caso de uso	Crear Tipo de Dedicación de Tiempo	
Descripción	Creación de Tipo de Dedicación de Tiempo	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario selecciona la opción Tipos de Dedicación de Tiempo del menú de Datos Iniciales.	2. El sistema muestra las opciones de Tipos de Dedicación de Tiempo y una lista de los registros existentes.
	3. El usuario hace clic en la opción Crear Nuevo Registro.	4. El sistema muestra la página de registro para el ingreso de los datos.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Tipos de Dedicación de Tiempo.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página actual sin realizar ninguna acción.	
Pos-Condiciones	El Tipo de Dedicación de Tiempo ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 13. Caso de uso - Crear Roles

Código	UC-005	
Nombre del caso de uso	Crear Roles	
Descripción	Creación de Roles	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Administración y da clic en la opción Roles del menú Datos Iniciales.	2. El sistema muestra las opciones de Roles y una lista de los registros existentes.
	3. El usuario selecciona la opción Crear Nuevo Registro.	4. El sistema muestra la página de registro para el ingreso de los datos.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Roles.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página actual sin realizar ninguna acción.	
Pos-Condiciones	El Rol ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 14. Caso de uso - Crear Usuarios

Código	UC-006	
Nombre del caso de uso	Crear Usuarios.	
Descripción	Creación de Usuarios.	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Administración y da clic en la opción Usuarios del menú de Datos Iniciales.	2. El sistema muestra las opciones de la Usuarios y una lista de los registros existentes.
	3. El usuario hace clic en la opción Crear Nuevo Registro.	4. El sistema muestra la página de registro para el ingreso de los datos correspondiente a la entidad usuario.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Usuarios.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página sin realizar ninguna acción.	
Pos-Condiciones	El Usuarios ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 15. Caso de uso - Asignar Roles a Usuarios

Código	UC-007	
Nombre del caso de uso	Asignar Roles a Usuarios	
Descripción	Asignación de Roles a Usuarios	
Actores	Administrador	
Pre-Condiciones	<p>Usuario correctamente autenticado y autorizado como administrador del sistema.</p> <p>Deben existir registros de usuarios.</p> <p>Deben existir registros de roles.</p>	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Administración y da clic en la opción Asignación de Roles del menú de Datos Iniciales.	2. El sistema muestra las opciones de Asignación de Roles y una lista de los usuarios y sus roles asignados.
	3. El usuario selecciona la opción Editar.	4. El sistema muestra la página para la edición de roles.
	5. El usuario modifica los datos y da clic en Grabar Roles.	6. El sistema registra los datos modificados y regresa a la página de Mantenimiento de Asignación de Roles.
Flujo Alternativo	<p>No existen registros de usuarios o roles, el sistema no permite la edición de roles.</p> <p>El usuario da clic en Regresar a la lista se cierra la ventana sin realizar ninguna acción.</p>	
Pos-Condiciones	El Rol del Usuario ha sido editado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 16. Caso de uso - Crear Tipos de Actividades

Código	UC-008	
Nombre del caso de uso	Crear Tipos de Actividades.	
Descripción	Creación de Tipos de Actividades.	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Administración y da clic en la opción Tipos de Actividades del menú Datos Iniciales.	2. El sistema muestra las opciones de Tipos de Actividades y una lista de los registros existentes.
	3. El usuario selecciona la opción Crear Nuevo Registro.	4. El sistema muestra la página de registro para el ingreso de los datos del nuevo tipo de Actividad.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Tipos de Actividades.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página sin realizar ninguna acción.	
Pos-Condiciones	El Tipo de Actividad ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 17. Caso de uso - Asignar Actividades a Docentes

Código	UC-009	
Nombre del caso de uso	Asignar Actividades a Docentes.	
Descripción	Asignación de Actividades a Docentes.	
Actores	Administrador	
Pre-Condiciones	<p>Usuario correctamente autenticado y autorizado como administrador del sistema.</p> <p>Deben existir registros de usuarios docentes y registros de tipos de actividades.</p>	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Administración y da clic en la opción Asignación de Actividades a Docentes del menú Datos Iniciales.	2. El sistema muestra las opciones de Asignación de Actividades a Docentes y una lista de los Docentes con las actividades asignadas.
	3. El usuario selecciona la opción Agregar.	4. El sistema muestra la página de registro para el ingreso de los datos de la actividad a asignar.
	5. El usuario ingresa los datos solicitados y da clic en Crear.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Asignación de Actividades a Docentes.
Flujo Alternativo	<p>No existen registros de docentes o tipos de actividades, el sistema no permite la asignación de actividades.</p> <p>El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error.</p> <p>El usuario da clic en Regresar a la lista se cierra la página sin realizar ninguna acción.</p>	
Pos-Condiciones	La asignación de la Actividad al Docente ha sido registrada en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 18. Caso de uso - Crear Datos de Facultades

Código	UC-010	
Nombre del caso de uso	Crear Datos de Facultades.	
Descripción	Creación de Datos de Facultades.	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Administración y da clic en la opción Datos de Facultades del menú Facultades y Carreras.	2. El sistema muestra las opciones de Datos de facultades y una lista de los registros existentes.
	3. El usuario selecciona la opción Crear Nuevo Registro.	4. El sistema muestra el formulario de registro para el ingreso de la información.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, registra la información y regresa a la página de Mantenimiento de Datos de Facultades.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página sin realizar ninguna acción.	
Pos-Condiciones	Los datos de Facultades han sido registrados en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 19. Caso de uso - Crear Datos de Carreras

Código	UC-011	
Nombre del caso de uso	Crear Datos de Carreras.	
Descripción	Creación de Datos de Carreras.	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Administración y da clic en la opción Datos de Carreras del menú Facultades y Carreras.	2. El sistema muestra las opciones de Datos de Carreras y una lista de los registros existentes.
	3. El usuario selecciona la opción Crear Nuevo Registro.	4. El sistema muestra la página de registro para el ingreso de los datos de la carrera.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Datos de Carreras.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página sin realizar ninguna acción.	
Pos-Condiciones	Los datos de Carreras han sido registrados en el sistema.	

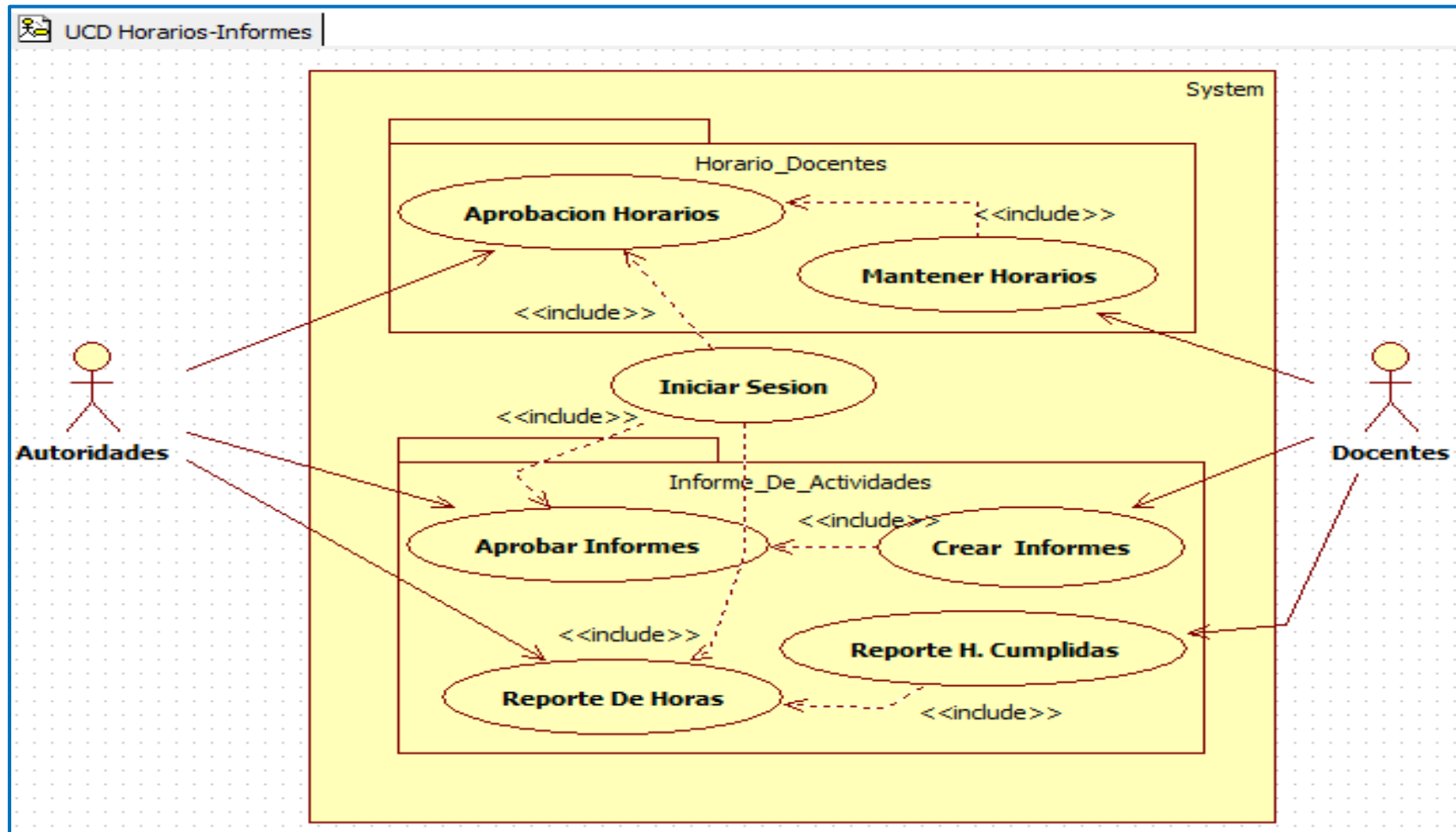
Fuente: Elaborado por el Autor

Cuadro 20. Caso de uso - Crear Períodos

Código	UC-012	
Nombre del caso de uso	Crear Periodo.	
Descripción	Creación de Periodo.	
Actores	Administrador	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como administrador del sistema.	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Administración y da clic en la opción Periodo del menú Periodos Lectivos.	2. El sistema muestra las opciones de Periodo y una lista de los registros existentes.
	3. El usuario selecciona la opción Crear Nuevo Registro.	4. El sistema muestra la página de registro para el ingreso de los datos del período.
	5. El usuario ingresa los datos solicitados y da clic en Grabar.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Periodo.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la página sin realizar ninguna acción.	
Pos-Condiciones	El nuevo Periodo ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Ilustración 15. Casos de Uso - Horarios e Informes



Fuente: Elaborado por el Autor

Cuadro 21. Caso de uso - Crear Horario de Actividades

Código	UC-013	
Nombre del caso de uso	Crear Horario de Actividades.	
Descripción	Creación de Horario de Actividades.	
Actores	Docentes	
Pre-Condiciones	Usuario correctamente autenticado y autorizado como Docente.	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Docentes y da clic en la opción Mantener Horarios de Actividades del menú Horarios de Actividades.	2. El sistema muestra la página de ingreso del horario de actividades.
	3. El usuario ingresa los datos solicitados y da clic en Crear Horario.	4. El sistema valida los datos, guarda la información y regresa a la página de Docentes.
Flujo Alternativo	El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error. El usuario da clic en Regresar a la lista se cierra la ventana sin realizar ninguna acción.	
Pos-Condiciones	El nuevo Horario de Actividades ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 22. Caso de uso - Crear Informe de Actividades

Código	UC-014	
Nombre del caso de uso	Crear Informes de Actividades.	
Descripción	Creación de Informes de Actividades.	
Actores	Docentes	
Pre-Condiciones	<p>Usuario correctamente autenticado y autorizado como Docente.</p> <p>El docente debe tener aprobado el horario de actividades.</p>	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Docentes y da clic en la opción Crear Informes del menú de Actividades.	2. El sistema muestra las opciones de Creación de Informes.
	3. El usuario selecciona el Periodo, el mes y el tipo de actividad.	4. El sistema muestra la página de registro para el ingreso de los datos de la actividad seleccionada.
	5. El usuario ingresa los datos solicitados y da clic en Grabar Informe.	6. El sistema valida los datos, guarda la información y regresa a la página de Mantenimiento de Informe de Actividades.
Flujo Alternativo	<p>El docente no tiene aprobado el horario de actividades, el sistema presenta un mensaje de advertencia.</p> <p>El sistema verifica la información, en caso de ser incorrecta presenta un mensaje de error.</p> <p>El usuario cancela el ingreso de Informes y regresa al menú principal de Docentes.</p>	
Pos-Condiciones	El Informe de Actividades ha sido registrado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 23. Caso de uso - Reporte de Horas Cumplidas

Código	UC-015	
Nombre del caso de uso	Reporte de Horas Cumplidas.	
Descripción	Generación de Reporte de Horas Cumplidas.	
Actores	Docentes	
Pre-Condiciones	<p>Usuario correctamente autenticado y autorizado como docente.</p> <p>El docente debe tener registros de informes de actividades.</p>	
Flujo Principal	<p>1. El usuario ingresa a la pantalla principal de Docentes y da clic en la opción Reporte de Horas Cumplidas del menú Informes de Actividades.</p>	<p>2. El sistema muestra las opciones de Reporte de Horas Cumplidas.</p>
	<p>3. El usuario selecciona el Periodo, el mes y selecciona la actividad y envía a generar el archivo.</p>	<p>4. El sistema genera el archivo en formato Excel y lo Descarga.</p>
Flujo Alternativo	<p>El sistema comprueba la existencia de la información, en caso de no existir presenta un mensaje de advertencia.</p> <p>El usuario no envía a generar el archivo en formato Excel y solo realiza la consulta en pantalla.</p>	
Pos-Condiciones	El archivo ha sido generado y descargado en formato Excel.	

Fuente: Elaborado por el Autor

Cuadro 24. Caso de uso - Aprobar Horario de Actividades

Código	UC-016	
Nombre del caso de uso	Aprobar Horario de Actividades.	
Descripción	Aprobación de Horario de Actividades.	
Actores	Autoridades	
Pre-Condiciones	<p>Usuario correctamente autenticado y autorizado como Coordinador.</p> <p>Deben existir uno o varios registros de horarios de actividades de los docentes.</p>	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Coordinadores y da clic en la opción Aprobar Horarios del menú Horario de Docentes.	2. El sistema muestra las opciones de Horarios de Actividades.
	3. El usuario selecciona el Docente.	4. El sistema muestra el formulario con el Horario registrado por el docente.
	5. El usuario da clic en Aprobar.	6. El sistema valida los datos, guarda la información y regresa al formulario de Aprobar Horario de Actividades.
Flujo Alternativo	No existen registros de horarios de actividades de los docentes, el sistema presenta un mensaje de advertencia.	
Pos-Condiciones	El Horario de Actividades ha sido Aprobado en el sistema.	

Fuente: Elaborado por el Autor

Cuadro 25. Caso de uso - Aprobar Informe de Actividades

Código	UC-017	
Nombre del caso de uso	Aprobar Informes de Actividades.	
Descripción	Aprobación de Informes de Actividades.	
Actores	Coordinadores de Carrera	
Pre-Condiciones	<p>Usuario correctamente autenticado y autorizado como Coordinador.</p> <p>Deben existir registros de informes de actividades de los docentes.</p>	
Flujo Principal	1. El usuario ingresa a la pantalla principal de Coordinadores de Carrera y da clic en la opción Aprobar Informes del menú Informes de Actividades.	2. El sistema muestra las opciones de Aprobación de Informes de Actividades.
	3. El usuario selecciona el Periodo, el mes y selecciona el tipo de actividad y da clic en buscar.	4. El sistema muestra los registros de Informes grabados por los docentes.
	5. El usuario selecciona el registro del informe de actividades que va a aprobar.	6. El sistema muestra los datos del Informe de Actividades seleccionado.
	7. El usuario da clic en la opción Aprobar Informe.	6. El sistema valida los datos, guarda la información y regresa al formulario de Aprobación de Informe de Actividades.
Flujo Alternativo	No existen registros de actividades confirmados, el sistema no procesa ninguna aprobación.	
Pos-Condiciones	El Informe de Actividades ha sido aprobado por el coordinador de carrera.	

Fuente: Elaborado por el Autor

Cuadro 26. Caso de uso - Reporte de Horas en Informes de Actividades

Código	UC-018	
Nombre del caso de uso	Reporte de Horas en Informes de Actividades.	
Descripción	Generación de Reporte de Horas en Informes de Actividades.	
Actores	Coordinadores de Carrera.	
Pre-Condiciones	<p>Usuario correctamente autenticado y autorizado como Coordinador.</p> <p>Deben existir registros de informes de actividades aprobados.</p>	
Flujo Principal	<p>1. El usuario ingresa a la pantalla principal de Coordinadores de Carrera y da clic en la opción Reporte de Horas del menú de Informes de Actividades.</p>	<p>2. El sistema muestra las opciones de Reporte de Horas en Informes de Actividades.</p>
	<p>3. El usuario selecciona el Periodo, el mes y selecciona la actividad y manda a generar el archivo.</p>	<p>4. El sistema genera el archivo en formato Excel y lo Descarga.</p>
Flujo Alternativo	<p>El sistema comprueba la existencia de la información, en caso de no existir presenta un mensaje de advertencia.</p> <p>El usuario no manda a generar el archivo en formato Excel y solo realiza la consulta en pantalla.</p>	
Pos-Condiciones	El archivo ha sido generado y descargado en formato Excel.	

Fuente: Elaborado por el Autor

4.6.2. Requerimientos No Funcionales

La aplicación web en general debe de tener un buen tiempo de respuesta, aunque esta característica depende de muchos factores como ancho de banda del servicio de internet que está usando el usuario, atenuación de la señal debido a factores climáticos, disponibilidad del servicio por parte del proveedor de internet, saturación del servidor de respuesta, entre otros.

El aplicativo web debe estar disponible las 24 horas del día asegurando la disponibilidad de la aplicación a todos los usuarios de cualquier nivel.

4.6.3. Requerimientos del sistema

El sistema debe de instalarse y publicarse en un equipo preferentemente con arquitectura de servidor, mismo que debe tener instalado el sistema operativo Windows versión para servidores. La base de datos a utilizarse en este aplicativo es Sql Server versión 2012.

4.6.4. Requerimientos de interfaz de usuario

La aplicación debe ejecutarse sobre la mayoría de exploradores más utilizados, para garantizar la compatibilidad con estos programas, utilizando estándares en el diseño de las páginas web, para evitar la sobrecarga de elementos innecesarios a fin de que el rendimiento sea aceptable y con interfaces amigable para el usuario.

4.6.5. Requerimientos de Seguridad

El software debe permitir un acceso al sistema con verificación de nombre de usuario y clave, que garantice la identidad y el rol de la persona que va a utilizar el aplicativo, esto permitirá que las opciones y el acceso a la información sea seccionada de acuerdo al perfil del usuario. El sistema también deberá implementar en su código las características de seguridad contra los ataques maliciosos por los hackers y evitar las técnicas de filtrado de información en las peticiones POST conocidas como OverPosting y la validación del token del usuario que inició la sesión en cada una de las llamadas HTTPPOST.

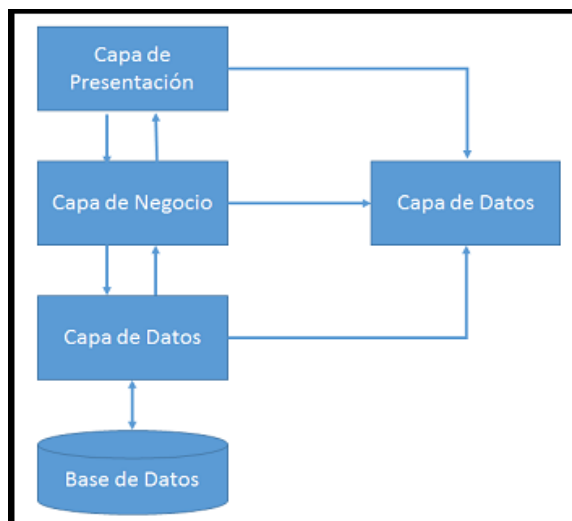
4.6.6. Especificaciones Técnicas

La aplicación fue diseñada bajo la arquitectura de cuatro capas, con el fin de mantener cada uno de los componentes con funciones específicas y el mantenimiento de la aplicación sea más fácil y organizado.

1. Capa de Entidades (Capa Transversal)
2. Capa de Negocio
3. Capa de Datos
4. Capa de Presentación

El desarrollo de una aplicación web en tres o más capas (Componentes) permite en primer lugar la reutilización de código en otros proyectos, el uso de una capa transversal en este caso la capa de entidades permite el correcto uso de las clases de dominio usadas en el modelo de desarrollo CodeFirst, de igual forma la capa de datos a través de la capa de entidades interactúa solo con la capa de negocio para facilitar el mantenimiento del código en casos donde la aplicación necesite trabajar con un motor de base de datos diferente y por último la capa de presentación utilizando sus interfaces de usuario (GUI) y por medio de las capa de negocio y entidades, realiza los procesos de solicitudes y actualización de información a la capa de datos, como se muestra en la siguiente figura:

Ilustración 16. Diagrama Arquitectura 4 Capas



Fuente: Elaborado por el Autor

Capa de Entidades:

Capa transversal disponible para todos los proyectos que conforman la solución, contiene las clases de dominio a ser utilizadas en el modelo de desarrollo Code First, mismas que servirán para que la capa de datos a través de una clase especial que implementa las clases DBContext y DBSet cree la base de datos en caso de no existir en el motor que se esté usando. Esta clase también actúa como un componente intermedio entre los componentes de la solución para permitir la manipulación de los objetos entidades ya sea para su consulta o escritura en el repositorio de datos, es utilizada a lo largo de toda la aplicación puesto que representa la única manera de transportar los datos de una capa a otra de una forma virtual. Esta capa también contiene dos interfaces que le permite a la capa de datos crear el repositorio de datos en memoria con las clases que implementen estas interfaces. Esta capa de entidades contiene las siguientes clases e interfaces:

Clases del Proyecto y de Dominio:

- Carrera
- Dedicacion
- Facultad
- RelaLabo
- Periodo
- Rol
- Tipolden
- TipoActividad
- Usuario
- UsuarioRol
- ActividadDocente
- Horario
- DetalleHorario
- MaestroFormato
- DetalleFormato

- MaestroInforme
- DetalleInforme
- AnioMes
- DetaFormTmp
- FechaHorarioHoras
- Login
- MaestroDetalleFormato
- PeriodosParaConsultas
- RolesAsignados
- RolesDelUsuario
- UsuarioListViewModel
- UsuarioParaConsultas
- UsuariosIndexData

Interfaces:

- IGenericRepository
- IUnitOfWork

Capa de Negocio:

Esta capa contiene las clases que interactúan con las clases de la capa de entidades, clases de la capa de presentación y clases de la capa de negocio. En esta capa existe una clase por cada una de las clases de la capa de presentación, pero interactúan de manera independiente es decir una clase de la capa de presentación puede acceder a cualquiera de las clases de la capa de negocio en caso de requerirlo, las clases de la capa de negocio contiene sus propiedades y métodos que atenderá los requerimientos de las clases de la capa de presentación para utilizar en sus procesos de paso y recepción de mensajes, variables y clases de la capa de entidades. Las clases que componen esta capa son las siguientes:

- LogicaActividadDocente
- LogicaCarrera
- LogicaDedicacion

- LogicaFacultad
- LogicaHorario
- LogicaInforme
- LogicaMaestroFormato
- LogicaPeriodo
- LogicaRelaLabo
- LogicaRol
- LogicaTipoActividad
- LogicaTipolden
- LogicaUsuario
- LogicaUsuarioRol

Capa de Datos:

Esta capa posee las clases que van a interactuar con las clases de la capa de negocio y las clases de la capa de entidades para los procesos de lectura y/o escritura en la base de datos en forma exclusiva, es decir es la única capa que se conecta con la base de datos. Esta esta capa posee dos clases especiales, una clase que implementa la interfaz IGenericRepository y una clase que implementa la interfaz IUnitOfWork y que también deriva de la clase DbContext utilizando la clase DBSet para crear el contexto de las entidades y así mismo crear el modelo de datos, mismo que servirá para crear la base de datos y las tablas en el sistema de base de datos en caso de que esta no exista, para este efecto en esta capa también encontramos la clase MyConfiguration que deriva de la clase DBConfiguration, misma que permite configurar el proveedor de servicio de la base de datos a utilizar y como va a responder la base de datos a las peticiones de lectura y escritura. Esta capa también contiene las clases utilizadas en los procesos de migración de la base de datos y que se crean cada vez que las clases de dominio cambian en la capa de entidades. Las clases que conforman esta capa son las siguientes:

- Configuration
- DatosActividadDocente
- DatosCarrera

- DatosDedicacion
- DatosFacultad
- DatosHorario
- DatosInforme
- DatosMaestroFormato
- DatosMaestroInforme
- DatosPeriodo
- DatosRelaLabo
- DatosRol
- DatosTipoActividad
- DatosTipolden
- DatosUsuario
- DatosUsuarioRol
- EfGenericRepository
- EfUnitOfWork
- ExpandoObjectExtension
- MyConfiguration

Capa de Presentación:

La capa de presentación o interfaz de usuario contiene las clase y vistas que se van a utilizar para interactuar con el usuario, quien a través de estas interfaces realizara las actividades de consultas, creación, edición y eliminación de datos, procesos más conocidos en el mundo de la informática como CRUD por sus siglas en inglés (CREATE, READ, UPDATE, DELETE), las clases de esta capa que pertenecen al grupo de controladores tienen la particularidad de que heredan de la clase BaseController, misma que a su vez hereda de la clase Controller, esto se hizo con el propósito de sobrescribir el método OnActionExecuting para controlar el tiempo de expiración de una sesión de usuario. En otro ámbito todas las clases usadas como controladores deben llevar el sufijo Controller. Las vistas de la capa de presentación han sido creadas implementando los plugins de javascript como son JQuery, Ajax, Json y las hojas de estilos presentes en el archivo Bootstrap.cs para estilizar las páginas web, a fin de presentar un entorno agradable a la vista del usuario,

dotando de cierta interacción de la aplicación en el lado del cliente (Explorador Web), los controladores y las vistas más importantes de esta capa son los siguientes:

Controladores:

- ActividadDocenteController
- BaseController
- DedicacionController
- FacultadController
- CarreraController
- HomeController
- HorarioController
- InformeController
- PeriodoController
- RelaLaboController
- ReportesController
- RolController
- TipoActividadController
- TipoldenController
- UsuarioController
- UsuarioRolController

Vistas:

- Home
Index
- Usuario
Agregar
- Horario
Crear
AutorizarHorario
- Periodo
Agregar
- Informe

CrearInforme
AprobarInforme
InformePdf

4.6.6.1. Estructura del modelo de datos:

El modelo aplicado para definir la estructura de datos es el relacional, con una base de datos normalizada en el mayor porcentaje posible a fin de garantizar una estructura confiable en cuanto a consistencia de la información, consultas relacionadas muy útiles en el modelo de desarrollo Code First, acceso dinámico a los datos lo cual ayuda al desarrollador a no tener que escribir más código en los métodos de acceso a información en consultas de tipo padre (Parent Rows) o de tipo hijo (Child Rows) mediante las relaciones lógicas definidas en la capa de entidades a través de las clases de dominio, y que se establecerán al momento de crear la base de datos con sus respectivas tablas.

A continuación se detalla la estructura de este modelo de datos aplicado con sus respectivas tablas.

Cuadro 27. Estructura Entidad Facultad

Entidad: Facultad		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
NombreFacultad	nvarchar(60)	Nombre de la facultad
UsuarioID	integer	Id del decano de la facultad

Fuente: Elaborado por el Autor

Almacena información relacionada a los datos de las facultades de la institución.

Cuadro 28. Estructura Entidad Carrera

Entidad: Carrera		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
FacultadID	integer	Id de la facultad a la que pertenece
NombreCarrera	nvarchar(60)	Nombre de la Carrera
UsuarioID	integer	Id del coordinador de carrera

Fuente: Elaborado por el Autor

Guarda información relacionada a los datos de las carreras de cada una de las facultades de la institución.

Cuadro 29. Estructura Entidad Tipolden

Entidad: Tipolden		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
TipoldenDescripcion	nvarchar(10)	Descripción del tipo de identificación

Fuente: Elaborado por el Autor

Aquí se registran los datos correspondientes a los tipos de identificación por ejemplo: Cédula, Pasaporte.

Cuadro 30. Estructura Entidad RelaLabo

Entidad: RelaLabo		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
RelaLaboDescripcion	nvarchar(10)	Descripción del tipo de relación laboral

Fuente: Elaborado por el Autor

Registran los datos relacionados a los tipos de relación laboral por ejemplo: Titular, Contratado.

Cuadro 31. Estructura Entidad Dedicacion

Entidad: Dedicacion		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
DediDescripcion	nvarchar(15)	Descripción del tipo de dedicación de tiempo

Fuente: Elaborado por el Autor

Almacena los datos correspondientes a los tipos de dedicación de tiempo de trabajo por ejemplo: Tiempo Completo, Parcial.

Cuadro 32. Estructura Entidad Rol

Entidad: Rol		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
Descripcion	nvarchar(20)	Descripción del tipo de rol

Fuente: Elaborado por el Autor

Guarda los datos relacionados a los tipos de rol del usuario por ejemplo: Docente, Coordinador, etc.

Cuadro 33. Estructura Entidad Usuario

Entidad: Usuario		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
TipoidenID	integer	Id del tipo de Identificación
Identificacion	nvarchar(20)	Número de identificación
NombreCorto	nvarchar(20)	Nombre corto (Inicio de sesión)
NombreCompleto	nvarchar(40)	Apellidos y nombres
TituloAcademico	nvarchar(60)	Titulo Academico
FechaNacimiento	datetime	Fecha de nacimiento
Nacionalidad	nvarchar(20)	Nacionalidad
Email	nvarchar(50)	Email
RelaLaboID	integer	Id del tipo de relación laboral
DedicacionID	integer	Id del tipo de dedicación de tiempo
Senia	nvarchar(200)	Seña para verificar clave del usuario
Contrasenia	nvarchar(200)	Contraseña para verificar clave del usuario

Fuente: Elaborado por el Autor

Registra los datos de los usuarios del sistema.

Cuadro 34. Estructura Entidad UsuarioRol

Entidad: UsuarioRol		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
UsuarioRolID	integer	Llave principal
UsuarioID	integer	Id del usuario
RolID	integer	Id del rol

Fuente: Elaborado por el Autor

Almacena la información de los roles asignados a los usuarios, por cada rol que se asigne a un usuario se crea un registro y por cada rol que se le quite a un usuario se elimina un registro.

Cuadro 35. Estructura Entidad TipoActividad

Entidad: TipoActividad		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal

DescripcionTipo	nvarchar(60)	Descripción del tipo de actividad
-----------------	--------------	-----------------------------------

Fuente: Elaborado por el Autor

Registra los datos de los tipos de actividades complementarias.

Cuadro 36. Estructura Entidad Periodo

Entidad: Periodo		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
AnioLectivo	nchar(9)	Año lectivo
FecIniClases	datetime	Fecha de inicio de clases
FecFinClases	datetime	Fecha de fin de clases
Ciclo	short integer	Primer ó Segundo (Enumeración)

Fuente: Elaborado por el Autor

Guarda los datos correspondientes a los periodos lectivos.

Cuadro 37. Estructura Entidad ActividadDocente

Entidad: ActividadDocente		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
TipoActividadID	integer	Id de la actividad
UsuarioID	integer	Id del docente
CarreraID	integer	Id de la carrera
PeriodoID	integer	Id del período académico
TotalHoras	integer	Número de horas a cumplir en la actividad

Fuente: Elaborado por el Autor

Registra los datos relacionados a las actividades complementarias asignadas a los docentes.

Cuadro 38. Estructura Entidad Horario

Entidad: Horario		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
UsuarioID	integer	Id del usuario
EstadoHorario	short integer	StandBy, Aprobado (Enumeración)

Fuente: Elaborado por el Autor

Almacena los datos correspondientes a los horarios de actividades creados por los docentes.

Cuadro 39. Estructura Entidad DetalleHorario

Entidad: DetalleHorario		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
TipoActividadID	integer	Id del tipo de actividad al que corresponde
HorarioID	integer	Id del horario al que corresponde
IniciaTerminaLunes	nvarchar(11)	Hora de inicio y fin de la actividad
IniciaTerminaMartes	nvarchar(11)	Hora de inicio y fin de la actividad
IniciaTerminaMiercoles	nvarchar(11)	Hora de inicio y fin de la actividad
IniciaTerminaJueves	nvarchar(11)	Hora de inicio y fin de la actividad
IniciaTerminaViernes	nvarchar(11)	Hora de inicio y fin de la actividad
IniciaTerminaSabado	nvarchar(11)	Hora de inicio y fin de la actividad
IniciaTerminaDomingo	nvarchar(11)	Hora de inicio y fin de la actividad

Fuente: Elaborado por el Autor

Guarda los datos del detalle de los horarios.

Cuadro 40. Estructura Entidad MaestroFormato

Entidad: MaestroFormato		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
TipoActividadID	integer	Id del tipo de actividad
Vigente	short integer	Si ó No (Enumeración)

Fuente: Elaborado por el Autor

Almacena los datos de la cabecera de los formatos de ingreso de informes de las actividades complementarias.

Cuadro 41. Estructura Entidad DetalleInforme

DetalleInforme		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
MaestroInformeID	integer	Id del maestro del informe
DetalleFormatoID	integer	Id para obtener la columna en DetalleFormato
Valor	nvarchar(max)	Valor almacenado
Fila	integer	Número de la fila en relación con el formato

Fuente: Elaborado por el Autor

Registra los datos del detalle del informe de actividades complementarias.

Cuadro 42. Estructura Entidad MaestroInforme

MaestroInforme		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
ActividadDocenteID	integer	Id de la actividad del docente al que corresponde el informe
MaestroFormatoID	integer	Id del formato al que corresponde el informe
AnioMes	nvarchar(7)	Año y mes del informe que esta registrando
FechaRegistro	datetime	Fecha de registro del informe por parte del docente
UsuarioAproboID	integer	Id del usuario que aprobó el informe
FechaAprobado	datetime	Fecha de aprobación del informe
EstadoInforme	short integer	Enviado, Aprobado, Rechazado (Enumeración)
HorasAplicadas	integer	Horas aplicadas en esta actividad según el horario
HorasAnioMes	integer	Horas que debe cumplir en el mes informado

Fuente: Elaborado por el Autor

Registra los datos correspondientes a la cabecera de los informes de actividades complementarias realizadas por el usuario docente.

Cuadro 43. Estructura Entidad DetalleFormato

Entidad: DetalleFormato		
NOMBRE DE CAMPO	TIPO DE DATOS	DESCRIPCION
ID	integer	Llave principal
MaestroFormatoID	integer	Id del MaestroFormato al que pertenece
Fila	integer	Número de fila
AltoFila	integer	Alto de la fila
Columnas	integer	Número de columnas a ocupar
Cabecera	nvarchar(max)	Cabecera
TipoDeDato	nvarchar(10)	Tipo de dato: Número, Fecha, Texto
AlignHoriz	nvarchar(15)	Alineación Horizontal
AlignVerti	nvarchar(10)	Alineación Vertical
Valor	nvarchar(max)	Dato que contiene el registro
PermiteFormato	short integer	Si habilita la característica de formateo del texto
Visible	short integer	No, Si (Enumeración)

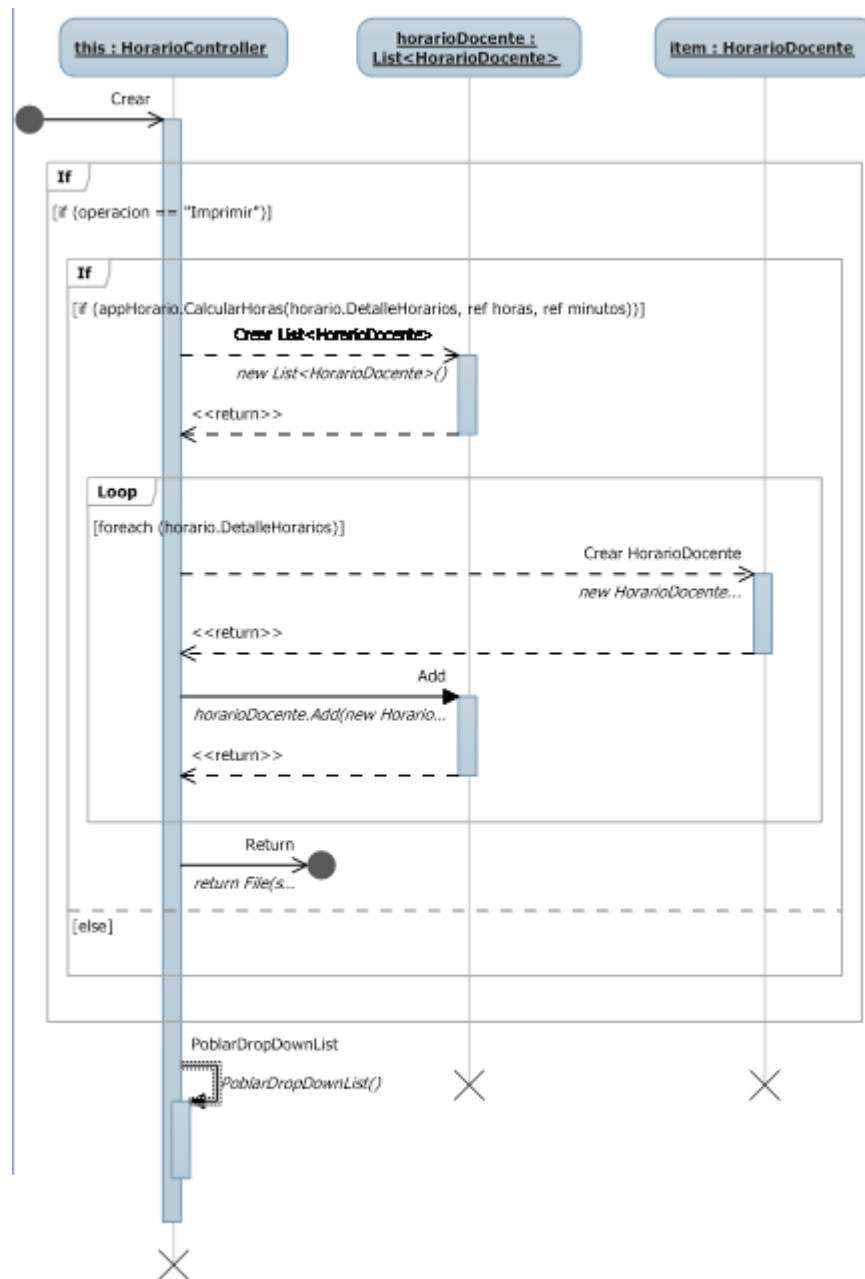
Fuente: Elaborado por el Autor

Almacena los datos relacionados al detalle de los formatos de ingreso de actividades complementarias.

4.6.6.2. Diagrama de Secuencia

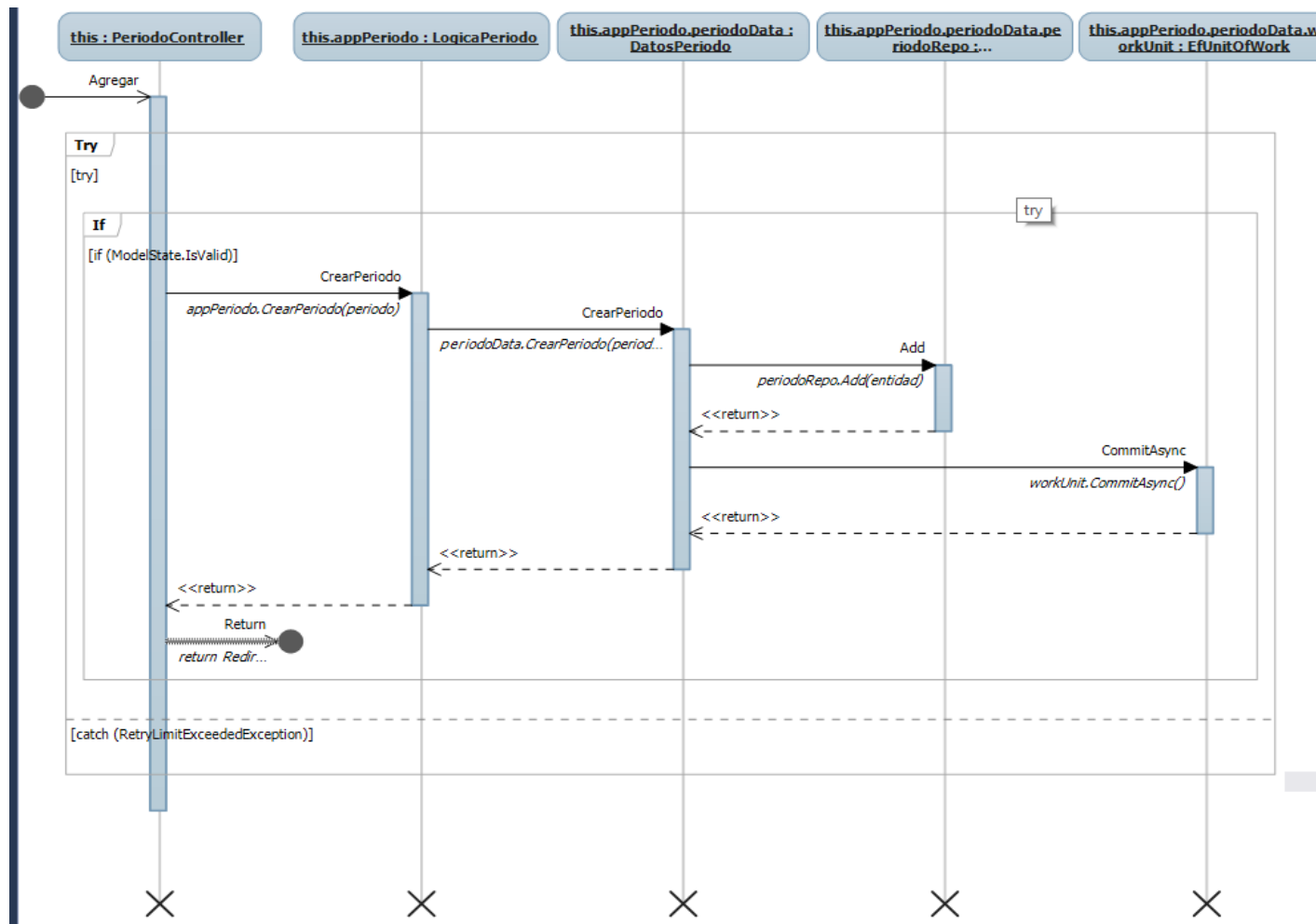
Mediante los diagramas de secuencia modelamos la interacción que existe entre los objetos del sistema a través del tiempo, dibujando los mensajes cronológicamente iniciando estos en la parte superior hacia la parte inferior.

Ilustración 17. Diagrama de Secuencia - Crear Horario



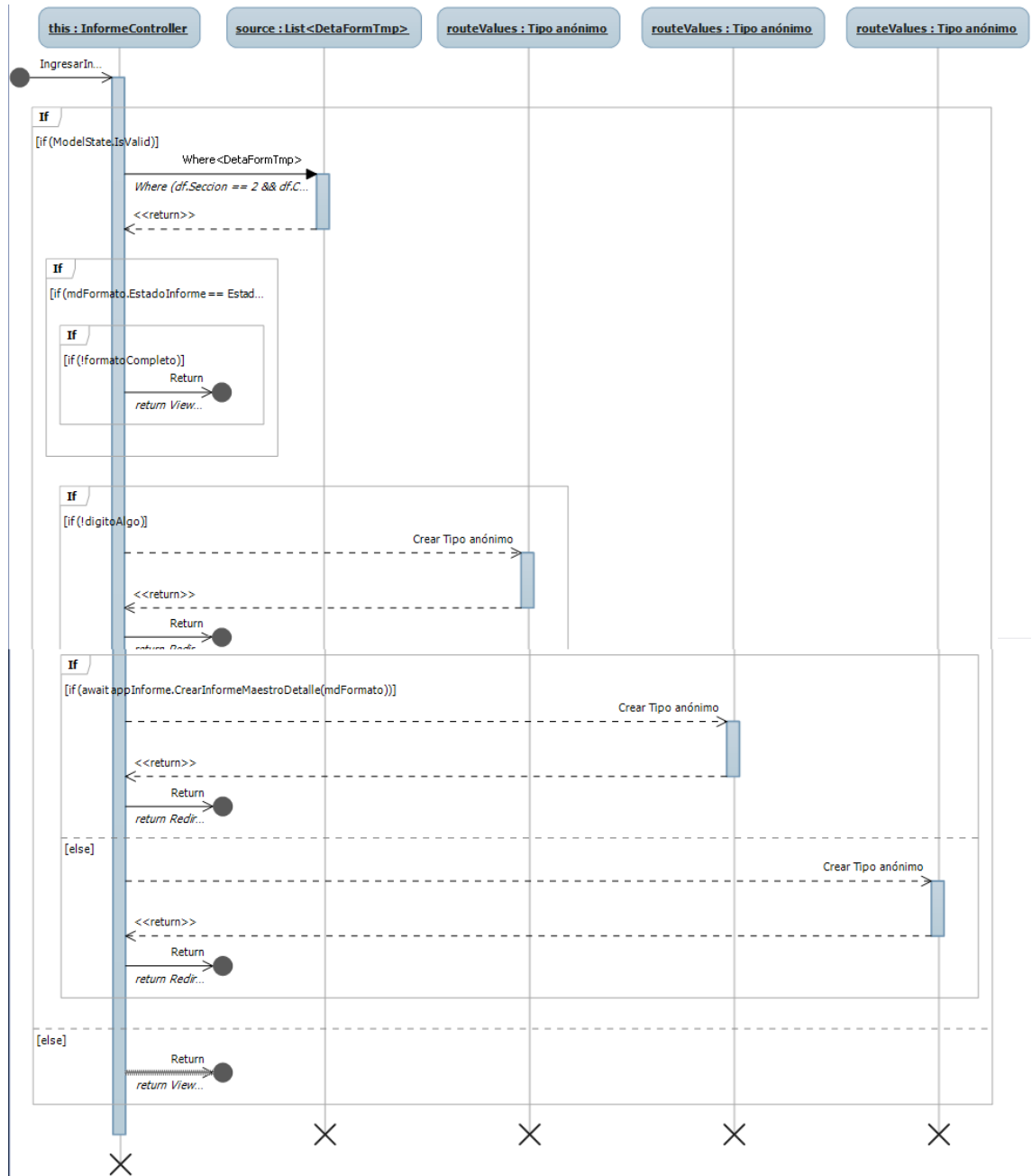
Fuente: Elaborado por el Autor

Ilustración 18. Diagrama de Secuencia - Crear Período



Fuente: Elaborado por el Autor

Ilustración 19. Diagrama de Secuencia - Crear Informe



Fuente: Elaborado por el Autor

4.6.6.3. Diagrama de Entidad - Relación

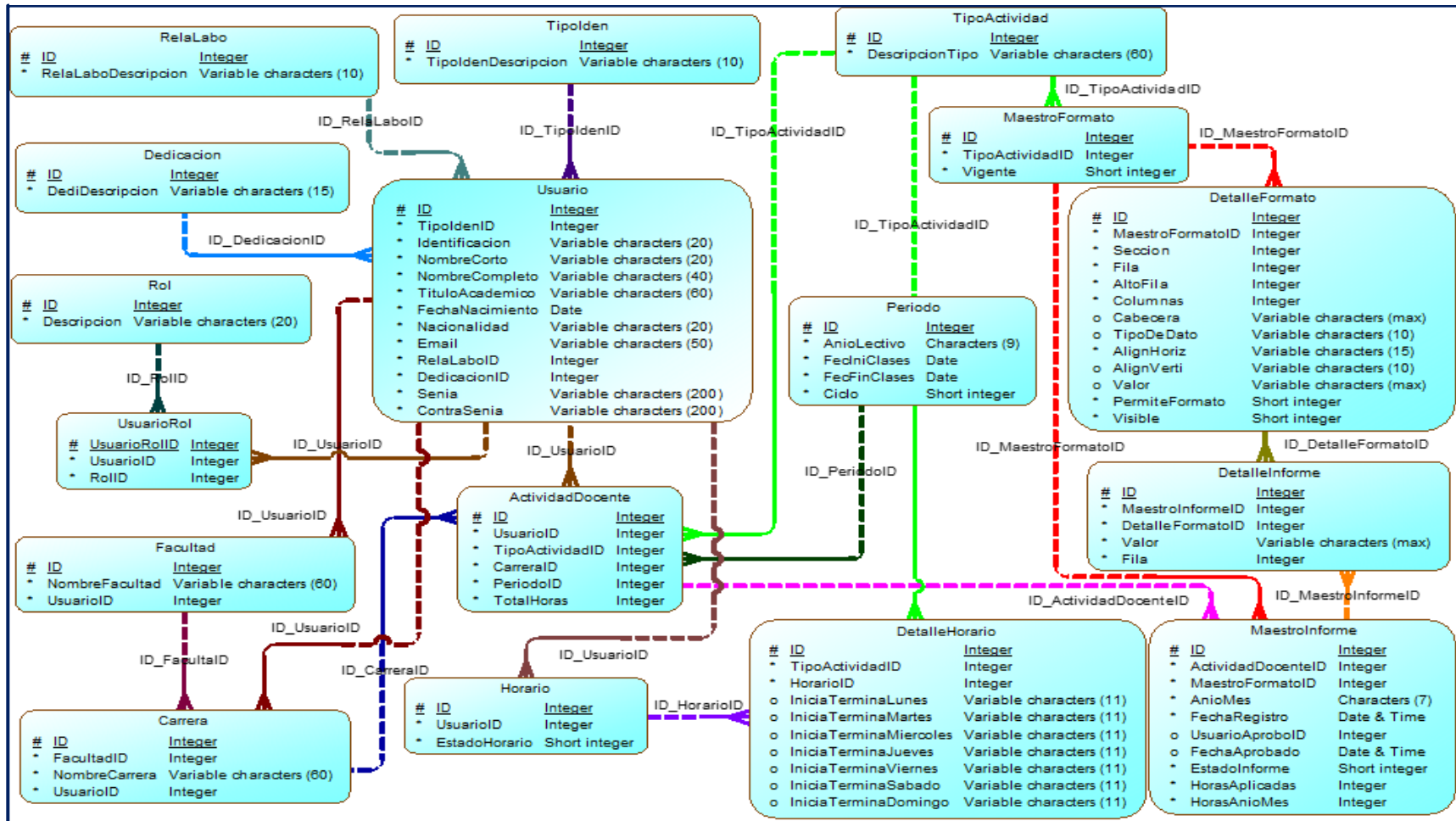
El modelo entidad relación compuesto por las entidades, que son objetos del mundo real y diferentes entre sí, descritos por medio de un conjunto de atributos, que a su vez agrupados en colecciones similares, conforman el conjunto de entidades con los mismos atributos y con su respectiva llave, referente a este modelo físico de datos plasmado en la base de datos, es fiel copia de las clases de dominio usadas en el modelo de desarrollo Code First como se indicó en el marco teórico.

Su diseño creado inicialmente en las clases de dominio cumple con los requerimientos según las necesidades especificadas y del tipo de información a almacenar, creándose las respectivas restricciones en las entidades relacionadas para especificar las llaves primarias y llaves foráneas.

Los accesos a la base de datos solo se permite a nivel de la aplicación, siempre y cuando los usuarios sean autenticados y autorizados, puesto que el software usa la autorización a través de formularios, por otra parte los datos referente a las claves de los usuarios han sido encriptados en dos campos especiales de la entidad usuario, a fin de proteger la información en caso de existir algún intento de inyección tipo sql.

En el siguiente diagrama Entidad-Relación, se indican cada una de las entidades (Tablas), creadas a través del proceso de inicialización de las clases especiales de la capa de datos, con sus llaves primarias (Atributos subrayados), y sus respectivas relaciones que indican las restricciones y dependencias en el tratamiento de los registros en los procesos de selección, creación, actualización y eliminación.

Ilustración 20. Diagrama de Base de Datos

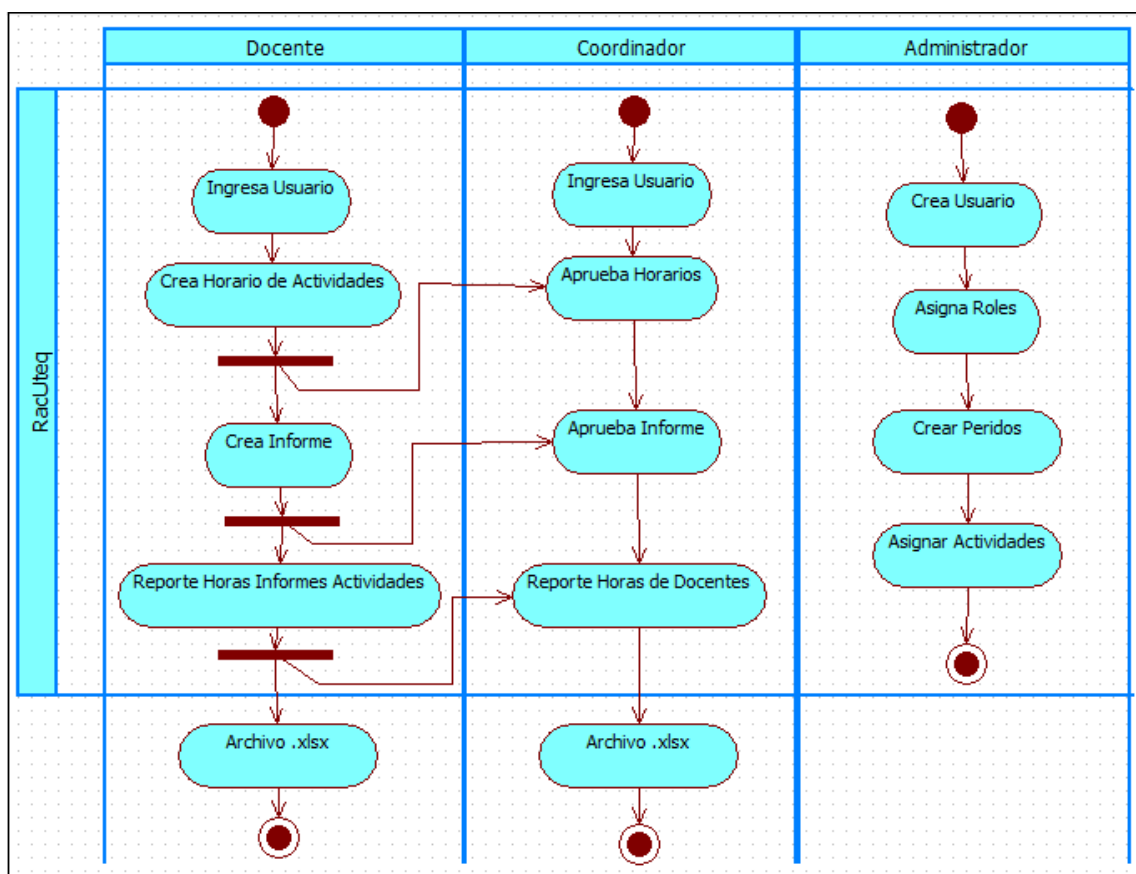


Fuente: Elaborado por el Autor

4.6.6.4. Diagrama de Actividades

Mediante el Diagrama de Actividades identificamos las interacciones entre los distintos tipos de usuarios del sistema, donde se indica la actividad que tendrá cada uno de ellos en el proceso de registro y control de actividades complementarias. Observamos las actividades preliminares del usuario Administrador en los procesos de inicialización y parametrización de los datos iniciales. El usuario docente inicia la actividad ingresando al sistema para luego crear su horario, crear sus informes y por último emitir el respectivo reporte. Las actividades de aprobación de horarios, aprobación de informes y emisión de reportes por parte de las autoridades son dependientes de las actividades que inicialmente realice el usuario docente.

Ilustración 21. Diagrama de Actividades

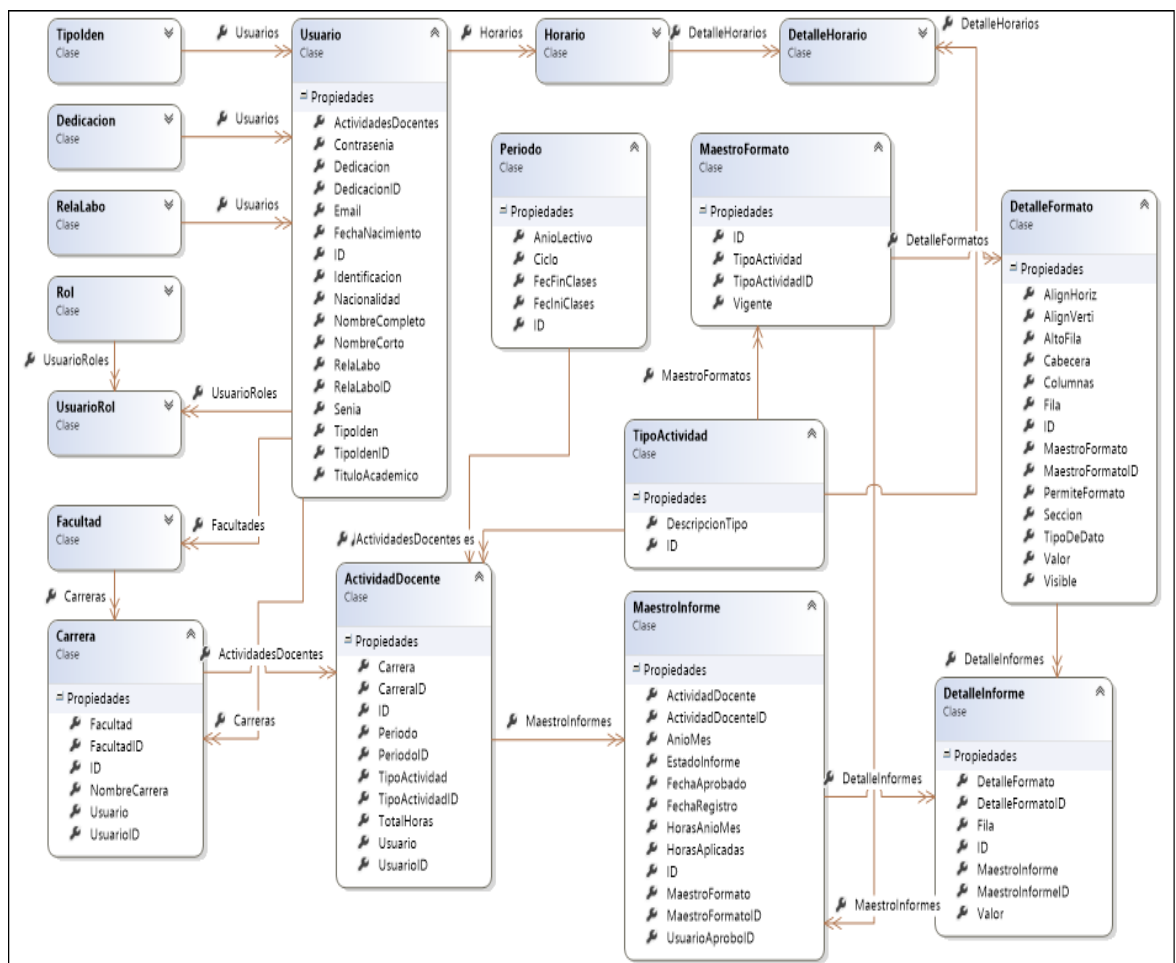


Fuente: Elaborado por el Autor

4.6.6.5. Diagrama de Clases

El Diagrama de Clases muestra las clases de dominio usadas en la capa de Entidades, para indicar las asociaciones que existen entre ellas en el manejo y control de colecciones cuando se apliquen las propiedades de navegación en los accesos al registro de la clase padre (Parent Rows) o a los registros de las clases hijas (Child Rows).

Ilustración 22. Diagrama de Clases



Fuente: Elaborado por el Autor

4.7. Interfaz de Usuario

4.7.1. Páginas Web Inicial y de Acceso al Sistema

La página web inicial que muestra cualquiera de los navegadores en internet es la siguiente, aquí el usuario podrá invocar a una pequeña pantalla (Modal) para poder ingresar al sistema.

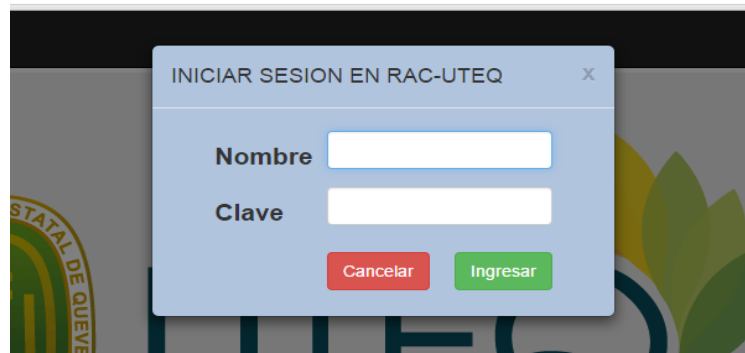
Ilustración 23. Página web inicial



Fuente: Elaborado por el Autor

Seleccionando la opción Iniciar Sesión el sistema mostrará la pantalla de inicio de sesión a continuación:

Ilustración 24. Pantalla de inicio de sesión

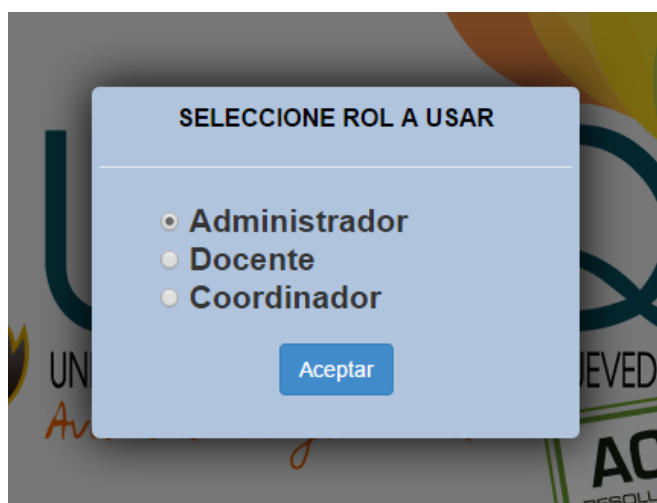


Fuente: Elaborado por el Autor

Para el efecto de ingresar al sistema como se indicó en el caso de uso anteriormente, el usuario debe digitar un nombre corto (Nick Name) y su respectiva clave, al ejecutar la opción Ingresar el sistema validará las credenciales ingresadas y a su vez permitirá o negará el ingreso al mismo.

Una vez que el usuario haya sido autenticado y autorizado por el sistema, si este tiene configurado más de un rol, es decir puede ser Administrador, Docente y Coordinador, el sistema exigirá al usuario que seleccione un tipo de rol con el cual va a trabajar en la aplicación.

Ilustración 25. Pantalla modal selección tipo de rol



Fuente: Elaborado por el Autor

4.7.2. Página Web del Menú del Administrador

Las siguientes pantallas corresponden a las páginas web de las opciones que tiene el usuario cuando haya ingresado con el rol de Administrador, bajo este perfil podrá ejecutar los siguientes procesos:

- Mantenimiento de Datos Iniciales
- Mantenimiento de Facultades y Carreras
- Mantenimiento de los Períodos Lectivos

Las opciones que corresponde a cada de una de las citadas anteriormente se muestran a continuación:

Ilustración 26. Opciones del menú de datos iniciales



Fuente: Elaborado por el Autor

Ilustración 27. Opciones del menú de facultades y carreras



Fuente: Elaborado por el Autor

Ilustración 28. Opciones del menú de períodos lectivos

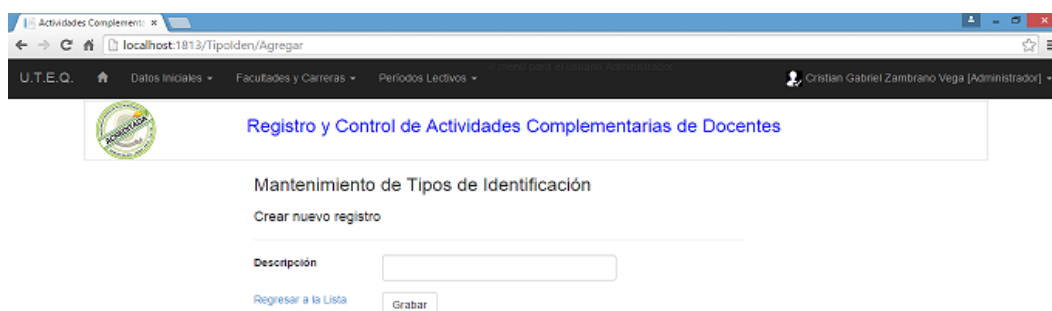


Fuente: Elaborado por el Autor

4.7.2.1. Página Web de Mantenimiento de Tipos de Identidades

En esta interfaz el usuario podrá crear los tipos de identificación necesarios para el mantenimiento de usuarios, ingresando los datos solicitados como se muestran en la siguiente pantalla:

Ilustración 29. Mantenimiento de tipos de identificación

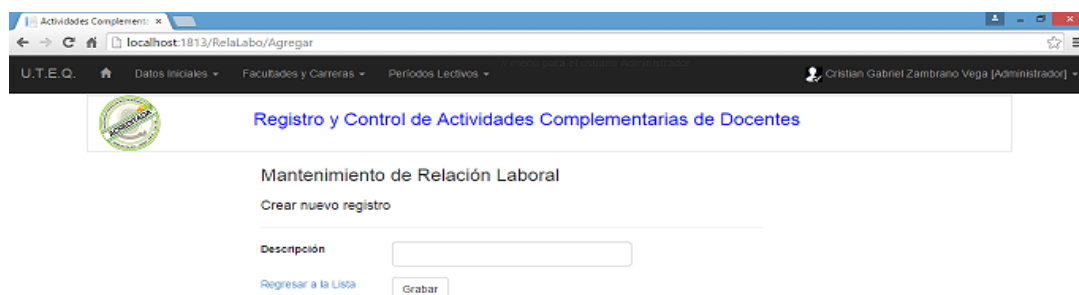


Fuente: Elaborado por el Autor

4.7.2.2. Página Web de Mantenimiento de Tipos de Relación Laboral

Interfaz que permite el ingreso de los tipos de relación Laboral necesarios para poder crear los usuarios.

Ilustración 30. Mantenimiento de tipos de relación laboral

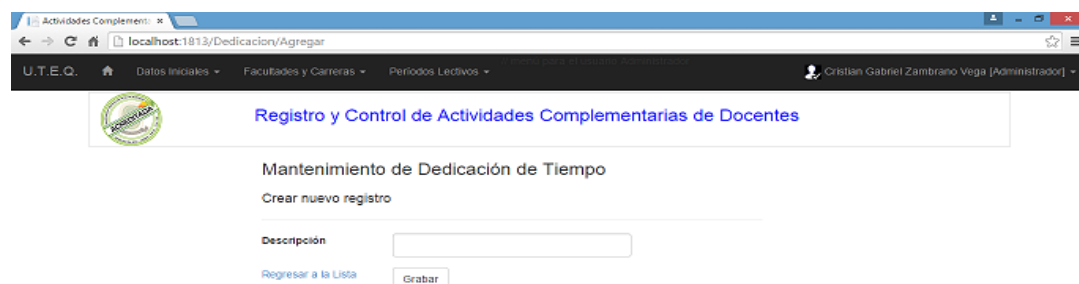


Fuente: Elaborado por el Autor

4.7.2.3. Página Web de Mantenimiento de Tipos de Tiempo Dedicado

Esta pantalla permite ingresar los distintos tipos de dedicación de tiempo de trabajo, por ejemplo: Tiempo Completo, Tiempo Parcial, entre otros.

Ilustración 31. Mantenimiento de tipos de dedicación de tiempo

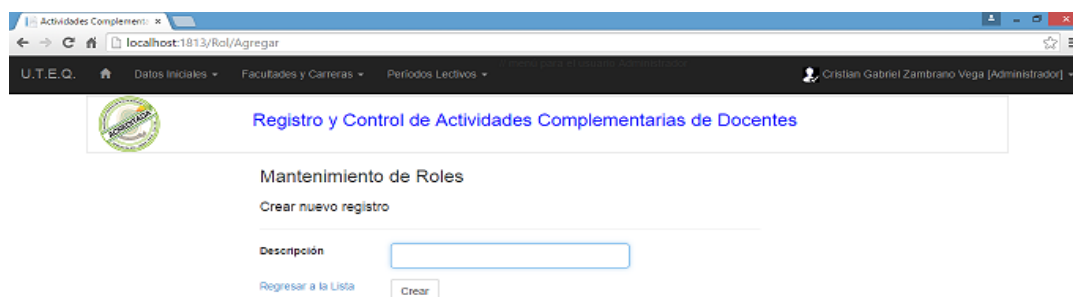


Fuente: Elaborado por el Autor

4.7.2.4. Página Web de Mantenimiento de Roles

Interfaz que permite dar mantenimiento a los tipos de roles a usarse en la aplicación, por ejemplo: Coordinador, Docente, etc.

Ilustración 32. Mantenimiento de tipos de roles



Fuente: Elaborado por el Autor

4.7.2.5. Página Web de Mantenimiento de Usuarios

Pantalla que permite la creación, edición y eliminación de usuarios del sistema, su interfaz es la siguiente:

Ilustración 33. Mantenimiento de usuarios



Fuente: Elaborado por el Autor

4.7.2.6. Página Web de Asignación de Roles a Usuarios

El usuario en esta pantalla podrá asignar los roles autorizados a cada uno de los usuarios.

Ilustración 34. Asignación de roles a usuarios

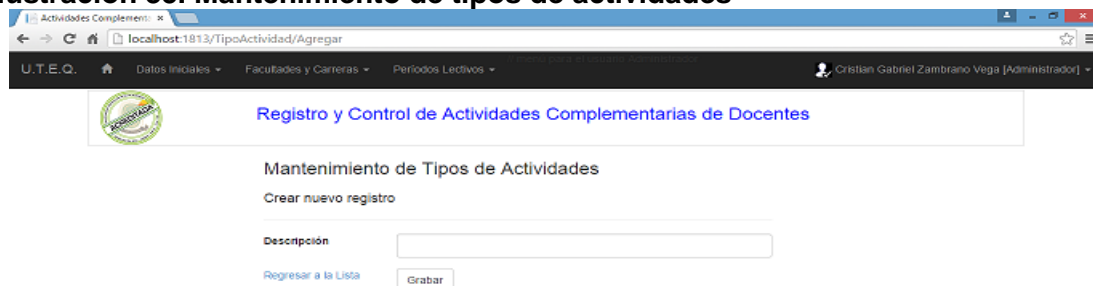


Fuente: Elaborado por el Autor

4.7.2.7. Página Web de Mantenimiento de Tipos de Actividades

Pantalla diseñada para que el Administrador pueda crear los Tipos de Actividades Complementarias.

Ilustración 35. Mantenimiento de tipos de actividades



Fuente: Elaborado por el Autor

4.7.2.8. Página Web de Asignación de Tipos de Actividades

Interfaz que permite la asignación de actividades complementarias a los docentes.

Ilustración 36. Asignación de tipos de actividades



The screenshot shows a web browser window with the URL `localhost:1813/ActividadDocente/Crear/1`. The page title is "Registro y Control de Actividades Complementarias de Docentes". Below the title, there is a section for "Mantenimiento de Actividades de Docentes" with a "Crear nuevo registro" link. The form contains the following fields:

Docente	Cristian Gabriel Zambrano Vega
Carrera	Ingeniería en Sistemas
Tipo de Actividad	Dirección de Tesis de Grado
Periodo	2015-2016 - Segundo
Total de Horas	120

There are also links for "Regresar a la Lista" and a "Crear" button.

Fuente: Elaborado por el Autor

4.7.2.9. Página Web de Mantenimiento de Facultades

Esta interfaz permite a los usuarios mantener los datos de las facultades, para solicitar los siguientes datos como se muestra en la figura a continuación:

Ilustración 37. Mantenimiento de Facultades



The screenshot shows a web browser window with the URL `localhost:1813/Facultad/Agregar`. The page title is "Registro y Control de Actividades Complementarias de Docentes". Below the title, there is a section for "Mantenimiento de Facultades" with a "Crear nuevo registro" link. The form contains the following fields:

Nombre Facultad	Facultad de Ciencias Agropecuarias
Autoridad a Cargo	Cristian Gabriel Zambrano Vega

There are also links for "Regresar a la Lista" and a "Grabar" button.

Fuente: Elaborado por el Autor

4.7.2.10. Página Web de Mantenimiento de Carreras

Esta pantalla sirve para mantener los datos de las carreras que pertenecen a cada una de las facultades de la institución:

Ilustración 38. Mantenimiento de datos de carreras

U.T.E.Q. Datos Iniciales Facultades y Carreras Períodos Lectivos Cristian Gabriel Zambrano Vega [Administrador]

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Carreras

Crear nuevo registro

Facultad: Unidad de Estudios a Distancia

Nombre Carrera: Ingenieria Comercial

Coordinador: Cristian Gabriel Zambrano Vega

Regresar a la Lista: Grabar

Fuente: Elaborado por el Autor

4.7.2.11. Página Web de Mantenimiento de Períodos Lectivos

Esta interfaz de usuario administrador, permite el mantenimiento de los períodos lectivos, como se muestra a continuación:

Ilustración 39. Mantenimiento de períodos lectivos

U.T.E.Q. Datos Iniciales Facultades y Carreras Períodos Lectivos Cristian Gabriel Zambrano Vega [Administrador]

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Períodos

Editar registro

Año Lectivo: 2015-2016

Fecha Inician Clases: 2015/05/15

Fecha Terminan Clases: 2015/10/30

Ciclo: Primero

Regresar a la Lista: Grabar

Fuente: Elaborado por el Autor

4.7.3. Página Web del Menú Docente

El usuario que se ingrese con el rol de Docente, podrá ejecutar los siguientes procesos:

- Mantenimiento de Horario de Actividades
- Mantenimiento de Informes de Actividades
- Consulta y/o Reportes de horas en actividades complementarias

4.7.3.1. Página Web de Mantenimiento de Horario de Actividades

Como podemos observar en la siguiente figura el usuario debe ingresar los datos correspondiente a las horas a utilizar en cada uno de los días de la semana para realizar sus respectivas actividades complementarias:

Ilustración 40. Mantenimiento de horario de actividades

U.T.E.Q. | Horario de Actividades | Informes de Actividades | Cristian Gabriel Zambrano Vega [Docente]

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Horario de Actividades

Estado:

Tipo de Actividad	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Dirección de Tesis de Grado	13:00-15:00						
Dirección de Tesis de Grado			15:00-17:00				
Dirección de Tesis de Grado					17:00-19:00		

Fuente: Elaborado por el Autor

4.7.3.2. Página Web de Mantenimiento Informes de Actividades

Esta interfaz sirve para dar mantenimiento a los Informes de Actividades Complementarias, en esta pantalla el usuario docente tiene la opción de poder salvar el informe como borrador para poder seguir editándolo o salvarlo como confirmado indicando que el informe está completo y listo para su aprobación por parte del coordinador de carrera.

Ilustración 41. Mantenimiento de informe de actividades

FECHA	HORARIO	HORAS	ACTIVIDAD REALIZADA	TAREAS ENCARGADAS	OBSERVACION
02/10/2015	17:00-19:00	2			
05/10/2015	13:00-15:00	2			
07/10/2015	15:00-17:00	2			
09/10/2015	17:00-19:00	2			
12/10/2015	13:00-15:00	2			
14/10/2015	15:00-17:00	2			
16/10/2015	17:00-19:00	2			

Fuente: Elaborado por el Autor

4.7.3.3. Página Web de Consulta y Reporte de Horas Cumplidas en actividades Complementarias

El usuario podrá en esta pantalla, consultar y/o emitir reportes de las horas cumplidas en el ejercicio de Actividades Complementarias, los reportes serán generados automáticamente en archivos con formato Excel (.xlsx).

Ilustración 42. Consulta y reporte de horas cumplidas

Registro y Control de Actividades Complementarias de Docentes

Reporte de Horas en Informes de Actividades

Periodo Lectivo: 2015-2016 - Primero Año-Mes: 2015-Mayo [Q] Todas las actividades

Fuente: Elaborado por el Autor

4.7.4. Página Web del Menú Coordinador

Las opciones disponibles para el usuario Coordinador son las siguientes:

- Aprobar Horario de Actividades
- Aprobar Informe de Actividades
- Reporte de Horas de Docentes

4.7.4.1. Página Web de Aprobación de Horarios a Docentes

Esta interfaz le permite al usuario coordinador ejecutar el proceso de Aprobación de Horarios a Docentes, para este efecto primero debe seleccionar el docente a tratar y a continuación el sistema mostrará el horario que el docente creó previamente y por último el coordinador ejecutará el proceso haciendo clic en la opción Aprobar Horario, la pantalla es la siguiente:

Ilustración 43. Aprobación de horario de actividades

Tipo de Actividad	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Dirección de Tesis de Grado	13:00-15:00						
Dirección de Tesis de Grado			15:00-17:00				
Dirección de Tesis de Grado					17:00-19:00		

Fuente: Elaborado por el Autor

4.7.4.2. Página Web de Aprobación de Informes de Actividades

Esta interfaz le permite al usuario coordinador ejecutar el proceso de Aprobación de Informes de Actividades, para lo cual primero debe consultar las actividades pendientes de aprobación y luego simplemente seleccionarla, revisar y aprobar, la pantalla es la siguiente:

Ilustración 44. Aprobación informe de actividades

Registro y Control de Actividades Complementarias de Docentes

Aprobación de Informes de Actividades

Periodo: 2015-2016 - Prime Año-Mes: 2015-Mayo Tipo Actividad: Dirección de Tesis de Grado

Fecha Registro	Nombre del Docente	Total Horas	Horas Aplicadas	Horas Incumplidas	Opciones
26/03/2015	Cristian Gabriel Zambrano Vega	14	4	10	Borrador

Fuente: Elaborado por el Autor

4.7.4.3. Página Web de Reporte de Horas de Docentes

Mediante esta pantalla podrá realizar consultas y/o reportes de horas aplicadas en Actividades Complementarias por parte de los docentes, la interfaz solicitará el período lectivo, el año y mes y el tipo de actividad si es necesario para generar la consulta o el reporte que será creado en un archivo con formato Excel (.xlsx), la interfaz es la siguiente:

Ilustración 45. Consulta y reporte de horas aplicadas

Registro y Control de Actividades Complementarias de Docentes

Reporte de Horas en Informes de Actividades

Periodo Lectivo: 2015-2016 - Primero Carrera: Ingeniería en Sistemas Año-Mes: 2015-Mayo Tipo Actividad: Todas las actividades

Fuente: Elaborado por el Autor

4.7.5. Test de satisfacción en el uso del software

El test de satisfacción se realizó mediante un simulacro que comprendió usuarios del nivel de docentes y coordinador de carrera, a fin de evaluar los aspectos principales de la aplicación web, quienes recibieron por parte del desarrollador del sistema la debida explicación de estas opciones y el flujo de actividades a seguir en el proceso de registro y control de actividades complementarias.

El conocimiento y la experiencia en el manejo de este tipo de aplicaciones por parte del personal permiten un correcto desempeño en la realización de este test, mismos que se indican en el siguiente cuadro.

4.7.5.1. Resumen del test de satisfacción en el uso del software

Cuadro 44. Resumen test de satisfacción en el uso del software

Módulo	Evaluadores	Horas de prueba	Solicitudes de cambio	Errores
Administración del Sistema	72	0.5	0	0
Acceso al sitio web	72	0.3	0	0
Crear Horario	72	0.5	0	0
Aprobar Horario	72	0.5	0	0
Crear Informe de Actividades	72	1	1	0
Aprobar Informes de Actividades	72	0.5	0	0
Consulta y Reporte de Horas del docente.	72	0.5	0	0
Consulta y Reporte de Horas de la carrera o escuela.	72	0.5	0	0

Fuente: Elaborado por el Autor

Como podemos observar el cuadro 44, el test fue muy satisfactorio y se cumplió con las especificaciones funcionales requeridas, mismo que se elaboró según las técnicas establecidas para los test de aceptación y certificación de sistemas informáticos, tabulado con la siguiente información:

- Nombres de los módulos,
- Número de evaluadores o personas elegidas para el test.
- Horas de prueba: Cálculo del tiempo que se tarda cada evaluador en revisar cada módulo.
- Solicitudes de cambio: Número total de propuestas expuestas a mejorar diferentes características del software como la interfaz gráfica, volumen de información, diseño, entre otros aspectos en la funcionalidad del software.
- Errores: Se registran algunas fallas que los usuarios evaluadores podrían tener al momento de testear la aplicación y que por algunos motivos durante el período de desarrollo no fueron detectadas, siendo un aporte importante por parte de los evaluadores quienes al no conocer en su totalidad las funciones del sistema encontrarían estas fallas.

En el módulo “Crear Informe de Actividades” se solicitó un cambio en la interfaz gráfica del ingreso de datos, el cual al no ser una modificación de tipo estructural sino más bien de forma, permitió una solución inmediata.

4.7.5.2. Permanencia del test por módulo

Cuadro 45. Tiempo de permanencia en el test por módulo

Módulo	Horas de prueba	%Duración
Administración del Sistema	0.5	11.63%
Acceso al sitio web	0.3	6.97%
Crear Horario	0.5	11.63%
Aprobar Horario	0.5	11.63%
Crear Informe de Actividades	1.0	23.25%

Aprobar Informe de Actividades	0.5	11.63%
Consulta y Reporte de Horas del docente.	0.5	11.63%
Consulta y Reporte de Horas de la carrera o escuela.	0.5	11.63%
Total Módulos: 8	4.3	100.00%

Fuente: Elaborado por el Autor

En base a los resultados del cuadro 45, el módulo que demandó mayor permanencia fue el correspondiente a: Crear Informe de Actividades, con el 23.25% del tiempo total con relación al resto de módulos. Es obvio que el tiempo de uso de este módulo debe ser mayor que el resto porque en esta interfaz los docentes ingresan los informes de actividades y el volumen de información es representativo y analítico, mientras que el módulo: Acceso al sitio web, fue el más bajo con un 6.97% debido a la facilidad de su uso.

4.7.5.3. Grado de aprobación por módulo

Cuadro 46. Grado de aprobación por módulo

Módulo	Evaluadores	Cambios solicitados	Aceptación
Administración del Sistema	72	0	100.00%
Acceso al sitio web	72	0	100.00%
Crear Horario	72	0	100.00%
Aprobar Horario	72	0	100.00%
Crear Informe de Actividades	72	1	98.61%
Aprobar Informe de Actividades	72	0	100.00%
Consulta y Reporte de Horas del docente.	72	0	100.00%
Consulta y Reporte de Horas de la carrera o escuela.	72	0	100.00%
Total Interacciones	626	1	99.84%

Fuente: Elaborado por el Autor

Los resultados del cuadro 46, muestran el gran nivel de aceptación de la aplicación, marcado por un 99.84%, existiendo solo una sugerencia de cambio que no implicó modificación alguna a nivel estructural, y que por su simplicidad fue solucionado en poco tiempo.

Como resultado de estas evaluaciones, se obtiene un sistema listo para ser implementado, con todas las funciones del diseño y que cuenta con una base sólida y escalable.

CAPITULO V
CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Como resultado de este trabajo de tesis, es posible concluir que:

- Se logra una mayor eficiencia en el control de las Actividades Complementarias de los Docentes, al reducir el tiempo en la elaboración de los informes generales de control por parte de las Unidades Académicas, además también se reduce el nivel de hojas impresas y así evitar el gasto en recursos materiales y perjuicio contra el medio ambiente.
- Se brinda mayor seguridad al proceso para evitar que la información sea extraviada durante las actividades de aprobación y obtención de firmas por parte de las autoridades de las Unidades Académicas, este proceso se realizaría a través del sistema informático en opciones especiales que garantizan la aprobación de los documentos electrónicos.
- Las Tecnologías de punta usadas en el desarrollo de esta aplicación web, brindan mayores ventajas y aumentan la producción en la creación de sistemas informáticos, que mantienen un control de calidad constante y una excelente agilización y efectividad en la automatización de procesos.

Con los resultados obtenidos por el desarrollo de una aplicación web previo a un estudio de los procesos tradicionales y semi-automatizados en el registro y control de actividades complementarias, y en base a las normas IEEE830, se puede concluir lo siguiente:

- El uso de herramientas para el diseño y lenguajes de programación de tecnología actualizada como visual studio 2013, motor de base de datos MS SQLServer 2012, marcos de trabajo como Entity Framework, .Net Framework, JavaScript, usadas en el desarrollo de la aplicación web para el proceso de control de la gestión de las actividades complementarias, permite incrementar la productividad en el desarrollo de software, en su mayoría son programas tipo open source de cuarta generación que facilitan la creación de sistemas informáticos escalables y de fácil mantenimiento.

- El sistema informático optimizó en gran escala los procesos de registro y control de las actividades complementarias, permitió el ahorro de tiempo, dinero, y el mejoramiento de la productividad del personal, para satisfacer los requerimientos y expectativas planteadas en la investigación.

5.2. Recomendaciones

- Se recomienda usar en la mayor escala posible la organización y los formatos propuestos en el estándar IEEE830, a fin de documentar debidamente los requerimientos para el desarrollo de la aplicación, tomando en consideración los diferentes puntos de vista entre los usuarios, cliente y equipo de desarrollo, para realizar los respectivos estudios de los procesos o metodologías a automatizar.
- Se debe usar software para el modelado y diseño de aplicaciones que permitan graficar estructura de datos, diagramas de clases, interacción entre usuario y sistemas, entre otros tipos de diagramas o ayudas gráficas que permiten tener un amplio conocimiento de la arquitectura de la aplicación a desarrollar, a fin de tener una mejor visión de los posibles problemas o mejoras en futuras versiones y también con el propósito de facilitar el mantenimiento ya sea por el mismo u otro equipo de desarrollo.
- Se recomienda también el uso de herramientas de desarrollo de software que vayan a la par con el avance tecnológico del hardware, teniendo en cuenta ciertas compatibilidades necesarias para el correcto funcionamiento con aplicaciones de terceros, necesarias en la interacción entre el usuario y el sistemas, por ejemplo los exploradores web en aplicaciones informáticas a través de internet.
- Se recomienda implementar la aplicación web en la Universidad Técnica Estatal de Quevedo, como un plan piloto en el primer semestre perteneciente al año de estudio 2015-2016, y luego en el segundo semestre del mismo período dejar establecida la implementación final en servidores netamente de producción, con algún tipo de mejora propuesto por los usuarios o departamentos de la institución vinculados con el control de la gestión de actividades complementarias.
- Se recomienda implementar posteriormente el uso de las firmas digitales, para que los coordinadores de carrera tenga un método más seguro de certificar la aprobación de los informes de actividades complementarias presentados por los docentes.

- Con el fin de garantizar un mejor rendimiento de la aplicación, se recomienda el uso de servidores de aplicación y servidores de datos indistintamente, para evitar la carga de trabajo, generalmente cuando se utiliza un solo equipo encargado de levantar y mantener varios servicios a la vez.
- Se recomienda que el registro de las actividades complementarias por parte de los docentes se ingrese semanalmente y a fin de mes se complemente el resto de información por ser este un informe de tipo mensual.
- Administrar las distintas opciones de la aplicación así como también un mantenimiento periódico a la Base de Datos a fin de garantizar su disponibilidad y consistencia de la información.
- Inducir a los usuarios de la aplicación las bondades del sistema y el correcto uso del mismo con el fin de obtener un mayor beneficio y optimización de las gestiones realizadas.
- Permitir las mejoras y actualizaciones de la aplicación, utilizar las herramientas de desarrollo de cuarta generación generalmente del tipo open source garantizando la escalabilidad del sistema.

CAPITULO VI
BIBLIOGRAFIA

6.1. Bibliografía

- Canós, Letelier y Penadés. (28 de 01 de 2015). *Metodologías Agiles en el desarrollo de software*. Obtenido de Universidad Politécnica de Valencia: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>. - 2003.
- Carles. (2014). *DESARROLLO DE APLICACIONES WEB*. ESPAÑA: EURECA MEDIA.
- Chadwick. (2011). *PROGRAMMING RAZOR*. ESTADOS UNIDOS: O'REILLY.
- Chaffer. (2013). *LEARNING JQUERY*. REINO UNIDO: OPEN SOURCE.
- Chatfield y Johnson. (2013). *Step by Step Microsoft Project 2013*. Estados Unidos: Microsoft.
- Crockford. (2008). *JAVASCRIPT THE GOOD PARTS*. ESTADOS UNIDOS: O'REILLY.
- Eguiluz. (2008). *INTRODUCCIÓN A AJAX*. ESPAÑA: CREATIVE COMMONS.
- Eguiluz. (2009). *INTRODUCCION A JAVASCRIPT*. ESPAÑA: CREATIVE COMMONS.
- Fitzgerald. (2012). *INTRODUCING REGULAR EXPRESSIONS*. ESTADOS UNIDOS: O'REILLY.
- Flores Muró, B., & Contreras Delgado, C. E. (2008). Modelo de investigación, aplicado en el desarrollo de Software. *Revistade estudios Politécnicos*, Vol VI, n. 9.
- Friedl. (2012). *MASTERING REGULAR EXPRESSIONS*. ESTADOS UNIDOS: O'REILLY.
- Gauchat. (2012). *EL GRAN LIBRO DE HTML5, CSS3 Y JAVASCRIPT*. ESPAÑA: MARCOMBO S.A.
- Guay. (2013). *BEGINNING ASP.NET MVC 4*. ESTADOS UNIDOS: APRESS.
- Haverbeke. (2011). *ELOQUENT JAVASCRIPT*. ESTADOS UNIDOS: AMAZON MEDIA.
- Howdle. (09 de 02 de 2012). *SMASHING MAGAZINE*. Obtenido de SMASHING MAGAZINE: <http://www.smashingmagazine.com/2012/02/09/beginners-guide-jquery-based-json-api-clients/>
- Javier Casares. (05 de 08 de 2011). *Keep It Simple Lab*. Obtenido de Keep It Simple Lab: <http://webperformanceoptimization.es/>
- Jeff. (2014). *VISUAL STUDIO 2013 COOKBOOK*. REINO UNIDO: PACK PUBLISHING LTD.

- Joyanes. (1998). *Programación Orientada a objetos*. Madrid: MCGRAW-HILL / Interamericana de España.
- Kendall. (2011). *Análisis y Diseño de Sistemas*. México: PEARSON EDUCACIÓN.
- Kniberg. (2007). *Scrum y XP desde las trincheras*. Estados Unidos: C4 Media Inc.
- Kniberg y Skarin. (2010). *Kanbas y Scrum - Obteniendo lo mejor de ambos*. Estados Unidos: C4Media Inc.
- Laboratorio Nacional de Calidad del Software. (2009). *INGENIERIA DEL SOFTWARE: METODOLOGIAS Y CICLO DE VIDA*. ESPAÑA: INTECO.
- Laudon. (2012). *Sistemas de Información Gerencial*. México: PEARSON EDUCACIÓN.
- Lerman. (1 de 12 de 2013). *MSDN Magazine*. Obtenido de MSDN Magazine: <https://msdn.microsoft.com/en-us/magazine/dn532202.aspx>
- Lerman y Miller. (2011). *Programming Entity Framework: Code First*. Estados Unidos: O'Reilly.
- Martinez. (24 de 10 de 2013). *Ventajas de las aplicaciones web*. Obtenido de www.MartinezAlegre.com:<http://www.martinezalegre.com/2013/10/ventajas-de-las-aplicaciones-web/>
- Mendez. (22 de 10 de 2008). *Universidad Complutense Informática*. Obtenido de Universidad Complutense Informática: <http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- Microsoft. (2012). *SQL Server 2012*. Estados Unidos: Microsoft Corporation.
- Microsoft. (21 de 01 de 2015). *Common Language Runtime*. Obtenido de MSDN Library: [https://msdn.microsoft.com/es-es/library/8bs2ecf4\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/8bs2ecf4(v=vs.110).aspx)
- Microsoft Corporation. (2014). *Microsoft Visual Studio Ultimate 2013*. Estados Unidos: Microsoft Corporation.
- MSDN. (21 de 01 de 2015). *Arquitectura ASP.NET*. Obtenido de [https://msdn.microsoft.com/es-es/library/aa719552\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa719552(v=vs.71).aspx)
- MSDN. (21 de 01 de 2015). *Lenguaje Visual C#*. Obtenido de [https://msdn.microsoft.com/es-es/library/aa287558\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa287558(v=vs.71).aspx)
- (OMG), O. M. (28 de 01 de 2015). *Lenguaje unificado de modelado*. Obtenido de http://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado
- Ramez y Shamkant. (2010). *FUNDAMENTALS OF DATABASE SYSTEMS*. ESTADOS UNIDOS: ADDISON-WESLEY.

- Revilla. (18 de 12 de 2013). *Las empresas sin sistemas automatizados tiene más problemas de rendimiento*: Obtenido de www.itespresso.es: <http://www.itespresso.es/empresas-sin-sistemas-automatizados-mas-problemas-rendimiento-119247.html>
- Rodriguez Roberto. (2011). *Programación Orientada a Objetos*. España: Creative Commons.
- UTEQ. (21 de 01 de 2015). *Universidad Técnica Estatal de Quevedo*. Obtenido de Nuestra Historia: <http://www.uteq.edu.ec/infopagina.php?recordID=33>
- UTEQ. (21 de 01 de 2015). *Universidad Técnica Estatal de Quvedo*. Obtenido de <http://www.uteq.edu.ec/infopagina.php?recordID=34>
- UTEQ-UPA. (15 de 01 de 2015). *Actividades Complementarias*. (W. Cevallos, Entrevistador)
- Wikipedia. (2012). *TWITTER BOOTSTRAP*. Obtenido de TWITTER BOOTSTRAP: http://es.wikipedia.org/wiki/Twitter_Bootstrap

CAPITULO VII
ANEXOS

7.1. Manual de Operación

7.1.1. Introducción

La aplicación web RAC-UTEQ es un software desarrollado para el control de la gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo. Mediante esta aplicación web, los docentes podrán llevar un control de los informes de actividades reportados a sus respectivos superiores (Coordinadores de Carrera), de igual las respectivas autoridades podrán verificar y avalorar la información registrada por parte de los docentes, con el fin de acreditar sus horas de trabajo consideradas como comisión en actividades complementarias.

El presente manual está dividido en tres secciones que explican las funciones para cada uno de los niveles de usuario siendo estos: Administrador, Docente y Autoridades, sin embargo la interfaz de inicio de sesión que es común a todos los usuarios sin importar el nivel de acceso, mediante la cual se realizará el proceso de autenticación y autorización, mostrará una pantalla modal adicional solamente a los usuarios que tengan asignado más de un rol, para que puedan elegir la jerarquía con la que realizarán sus operaciones durante la sesión de trabajo en el sistema.

Ingresar al Sistema (pantalla Inicial):

Después de llamar a la dirección web de la aplicación RAC-UTEQ, esta muestra su pantalla inicial.



Vista Iniciar sesión en RAC-UTEQ.



Pantalla modal de autenticación de usuario.

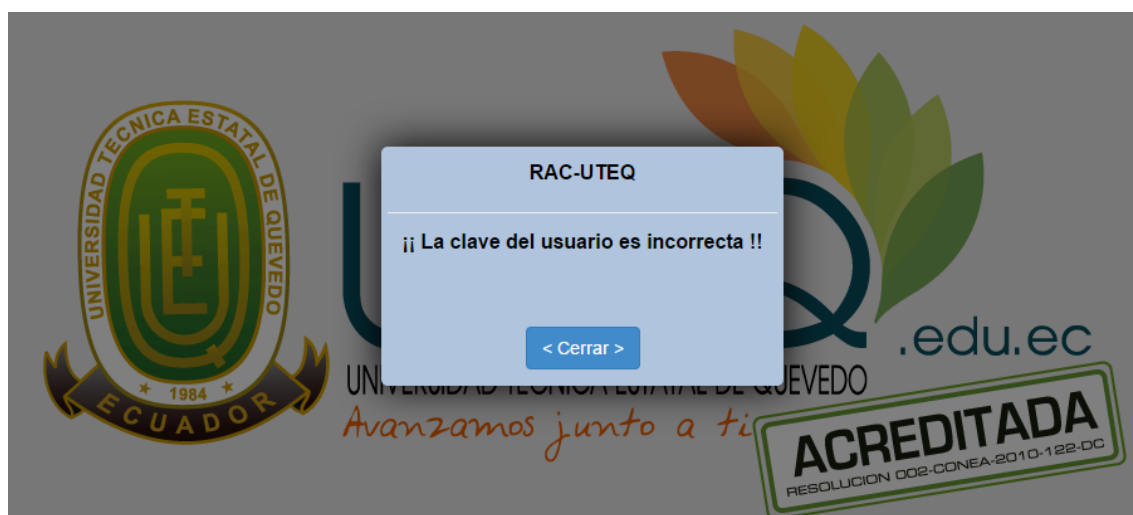
Se deben ingresar los datos solicitados y luego dar clic en aceptar o cancelar para cerrar la actual pantalla modal.

Si el usuario no existe, el sistema mostrará un mensaje de advertencia en una nueva pantalla modal, como se indica en la siguiente figura.



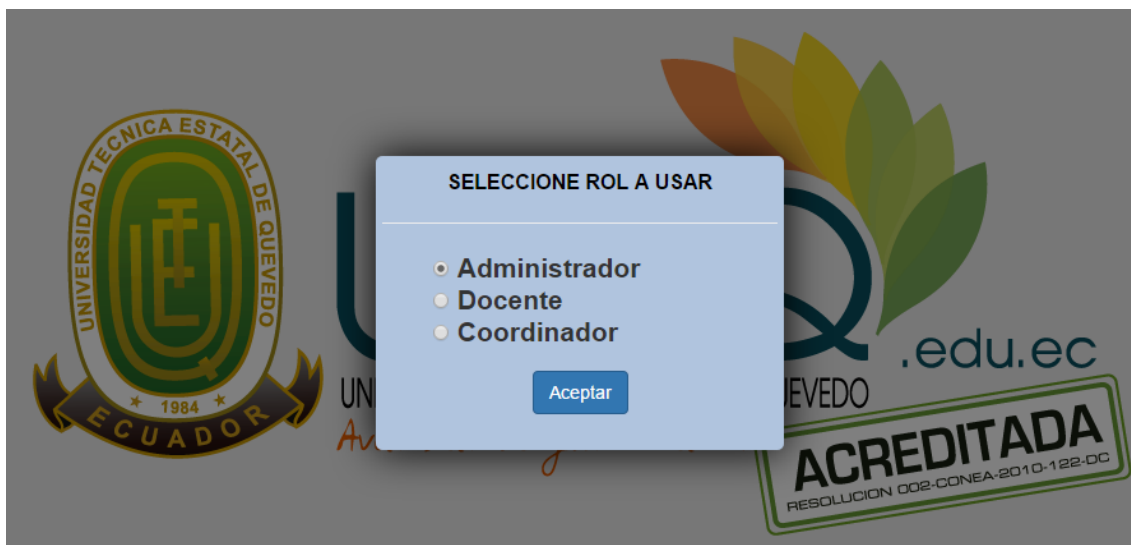
Pantalla modal de mensaje indicando que el usuario no existe.

De igual forma si el usuario existe pero la clave digitada es incorrecta el sistema mostrara otro mensaje de advertencia indicando el problema.



Pantalla modal de mensaje indicando que la clave no es válida.

Si las credenciales del usuario son correctas, es decir el usuario existe como tal y la clave digitada es correcta, el sistema mostrará a continuación una pantalla modal en caso de que este tenga asignado más de un rol, para que seleccione el rol según la función a realizar en la aplicación, como se muestra en la siguiente figura.



Pantalla modal para selección del rol a usar.

La figura anterior presenta una interfaz muy obvia, aquí simplemente el usuario debe seleccionar el tipo de rol con el que va a trabajar y que determinará las opciones a las que tendrá acceso según este nivel de usuario.


La interfaz gráfica en la siguiente figura muestra las opciones del usuario administrador:



Opciones del usuario con el rol de Administrador.

El menú de datos iniciales de las opciones del Administrador, sirve para crear los datos básicos que necesita el sistema para estar operativo en los procesos de registro de informes de actividades por parte del usuario docente, a continuación se indican las interfaces de los procesos de creación en cada una de las opciones del menú de datos iniciales:

Crear Tipos de Identificación (Cédula, Pasaporte, etc.):



Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Tipos de Identificación

Crear nuevo registro

Descripción

[Regresar a la Lista](#)

Digitar descripción del tipo de identificación

Clic aquí para grabar registro

Vista Crear Tipos de Identificación

Crear Tipos de relación Laboral (Titular, Contratado, etc.):



Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Relación Laboral

Crear nuevo registro

Descripción


[Regresar a la Lista](#)

Digitar descripción del tipo de Relación laboral

Clic aquí para grabar registro

Vista Crear Tipos de Relación Laboral

Crear Tipos de Tiempo Dedicado (Completo, Parcial, etc.):



Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Dedicación de Tiempo

Crear nuevo registro

Descripción

[Regresar a la Lista](#)

Digitar descripción del tipo de Tiempo dedicado

Clic aquí para grabar registro

Vista Crear Tipos de Dedicación de Tiempo

Crear Tipos de Roles (Administrador, Docente, etc.):

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Roles

Crear nuevo registro

Descripción

Regresar a la Lista

Digitar descripción del tipo de Rol

Clic aquí para grabar registro

Vista Crear Tipos de Roles

Crear Usuarios:

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Usuarios

Crear nuevo registro

Tipo Identificación

Nombre Usuario

Fecha Nacimiento

Nacionalidad

Relación Laboral

Clave

Regresar a la Lista

Identificación

Nombre Completo

Título Académico

Email

Tiempo Dedicado

Confirmar Clave

Ingresar datos requeridos

Clic para grabar el registro de usuario

Vista Crear Usuarios

En la interfaz de crear Usuarios, la aplicación exige que ingresemos todos los datos solicitados de:

Identificación.- Depende del tipo de identificación (cédula, pasaporte).

Nombre Usuario.- Un nombre pequeño para usar como usuario de inicio de sesión.

Nombre Completo.- Se debe digitar los dos nombres y los dos apellidos.

Fecha Nacimiento.- Ingresar la fecha de nacimiento del usuario a crear.

Título Académico.- Aquí se debe ingresar el título académico del usuario a crear.

Nacionalidad.- Nacionalidad del usuario (ecuatoriana, cubana, etc.).

Email.- Dirección del correo electrónico del usuario.

Relación Laboral.- Seleccionar el tipo de relación laboral.

Tiempo Dedicado.- Seleccionar el tipo de dedicación de tiempo de trabajo.

Clave y Confirmar Clave.- Datos equivalentes que le permitirán al usuario iniciar sesión en el sistema.

Asignación de Roles:

Esta interfaz es muy intuitiva, solo se requiere dar clic en los roles que se desea asignar y por último confirmar el registro de estos dando clic en el botón grabar.

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Roles de Usuarios

Asignar Roles

Usuario a Editar Cristian Gabriel Zambrano Vega

Administrador 6 Coordinador 3 Decano

7 Docente 4 Subdecano 5 Upa

2 Vicerrector

[Regresar a la Lista](#)

Clic para asignar o quitar roles

Clic para grabar roles

Vista Asignación de Roles a Usuario

Crear Tipos de Actividades:

Esta interfaz permite crear los tipos de actividades complementarias que posteriormente se asignarán a los usuarios docentes.

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Tipos de Actividades

Crear nuevo registro

Descripción

[Regresar a la Lista](#)

Digitar descripción del tipo de Actividad

Clic aquí para grabar registro

Vista Crear Tipo de Actividad

Asignación de Tipos de Actividades a Docentes:

Esta interfaz muestra los elementos necesarios que se deben seleccionar y digitar para asignar una actividad al docente:

Docente.- Seleccionar el docente de la lista.

Carrera.- Seleccionar la carrera de la lista.

Tipo de Actividad.- Seleccionar el tipo de actividad de la lista.

Período.- Seleccionar el período de la lista.

Total de Horas.- Digitar el número de horas a asignar durante el período.

The screenshot shows a web interface titled "Registro y Control de Actividades Complementarias de Docentes". Under the heading "Mantenimiento de Actividades de Docentes", there is a "Crear nuevo registro" section. The form includes the following fields: "Docente" (Cristian Gabriel Zambrano Vega), "Carrera" (Ingeniería en Sistemas), "Tipo de Actividad" (Dirección de Tesis de Grado), "Periodo" (2015-2016 - Primero), and "Total de Horas" (80). A "Crear" button is at the bottom right. A green oval highlights the form fields with arrows pointing to the text "Ingresar datos requeridos" and "Clic para crear asignación de actividad". A blue link "Regresar a la Lista" is located below the form.

Vista Asignación de Tipos de Actividades a Docentes

Crear Facultad:

Interfaz que permite crear un registro de una facultad.

The screenshot shows a web interface titled "Registro y Control de Actividades Complementarias de Docentes". Under the heading "Mantenimiento de Facultades", there is a "Crear nuevo registro" section. The form includes the following fields: "Nombre Facultad" (Facultad de Ciencias Agropecuarias), "Autoridad a Cargo" (Cristian Gabriel Zambrano Vega), and a "Grabar" button. A green oval highlights the form fields with arrows pointing to the text "Digitar nombre de facultad", "Seleccionar usuario autoridad", and "Clic aquí para grabar registro". A blue link "Regresar a la Lista" is located below the form.

Vista Crear Facultad

Crear Carrera:

Por medio de esta interfaz creamos un registro de una carrera.

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Carreras

Crear nuevo registro

Facultad: Unidad de Estudios a Distancia

Nombre Carrera: Ingeniería en Sistemas de Información

Coordinador: Cristian Gabriel Zambrano Vega

Regresar a la Lista

Grabar

Seleccionar Facultad

Ingresar Nombre de carrera

Seleccionar usuario coordinador

Clic para grabar

Vista Crear Carrera

Crear Período:

Esta vista permite crear un registro correspondiente a un período de clases.

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Períodos

Crear nuevo registro

Año Lectivo: 2015-2016

Fecha Inician Clases: 2015/05/09

Fecha Terminan Clases: 2016/10/23

Ciclo: Primero

Regresar a la Lista

Grabar

Ingresar año lectivo separado por un guion

Seleccionar fecha de inicio de clases

Seleccionar fecha de finalización de clases

Seleccionar ciclo

Clic para grabar

Vista Crear Período de Clases

Opciones del menú del usuario Docente:

Horario de Actividades ▾

Mantener Horario de Actividades

Informes de Actividades ▾

Crear Informes

Consulta y Edición de Informes

Reporte de Horas Cumplidas

Opciones del menú del usuario Docente

La figura anterior muestra las opciones del menú correspondiente al usuario Docente. Las siguientes imágenes indican las interfaces de estas opciones.

Crear Horario de Actividades:

La siguiente interfaz le permite al docente crear su horario de actividades, seleccionando e ingresando los siguientes datos:

Tipos de Actividad.- Debe seleccionar el tipo de actividad de la lista.

Hora Inicial y Final.- Cualquiera de los días que seleccione debe ingresar la hora inicial y final en el formato HH:mm-HH:mm, por ejemplo: 15:00-16:00.

Botón Agregar Actividad.- Permite crear un nuevo registro de horario.

Botón Eliminar Actividad.- Elimina la última actividad agregada.

Botón Imprimir Horario.- Imprime el horario en base a la última actualización del mismo, no los datos que aún permanecen en memoria y no han sido grabados.

Botón Crear Horario.- Actualiza los datos del horario que se está creando o editando.

Registro y Control de Actividades Complementarias de Docentes

Mantenimiento de Horario de Actividades

Estado: StandBy

Buttons: Agregar Actividad, Eliminar Actividad, Imprimir Horario

Tipo de Actividad	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Dirección de Tesis de Grado	13:00-15:00						
Dirección de Tesis de Grado			15:00-17:00				
Dirección de Tesis de Grado					17:00-19:00		

Crear Horario

Vista Crear Horario de Actividades

Crear Informe de Actividades:

Al seleccionar la opción de Crear Informes, el sistema muestra una interfaz que permite seleccionar el período de clases, el año y mes a procesar y el tipo de actividad que se va a informar, como se muestra en la siguiente figura:

Registro y Control de Actividades Complementarias de Docentes

Crear Informe de Actividades

PERIODO 2015-2016 - Primero 🔍 AÑO-MES 2015-Mayo ▼

Actividades asignadas a su cargo

Dirección de Tesis de Grado

Proyectos de Investigación Científica

Vista Selección del período, año y mes y tipo de actividad a informar del docente autenticado y autorizado.

La siguiente figura muestra la interfaz para crear el informe de actividades de un docente, después de que el usuario selecciona los parámetros indicados anteriormente. En esta vista debemos ingresar y seleccionar los siguientes elementos:

Fecha de registro.- Seleccionar o digitar la fecha de registro del informe.

Datos del Informe.- Ingresar los datos del informe de actividades según el formato preestablecido para el tipo de actividad a ingresar.

Estado Informe.- Seleccionar Borrador si el informe aún no está completo o Confirmado para que pueda ser aprobado por el coordinador de carrera.

Botón Grabar Informe.- Registra los datos del informe, si el estado es Borrador el informe puede ser editado nuevamente, pero si el estado es Confirmado, el informe queda bloqueado y solo puede ser leído o impreso.

UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO
INFORME DE ACTIVIDADES COMPLEMENTARIAS

FACULTAD: Unidad de Estudios a Distancia CARRERA: Ingenieria en Sistemas

ACTIVIDAD: Dirección de Tesis de Grado

DOCENTE: Cristian Gabriel Zambrano Vega PERIODO: 2015-2016-Primero Fecha: 04/04/2015

FECHA	HORARIO	HORAS	ACTIVIDAD REALIZADA	TAREAS ENCARGADAS	OBSERVACION
02/10/2015	17:00-19:00	2			
05/10/2015	13:00-15:00	2			
07/10/2015	15:00-17:00	2			
09/10/2015	17:00-19:00	2			
12/10/2015	13:00-15:00	2			
14/10/2015	15:00-17:00	2			

Ing. Dominga Ing. Cristian Zambrano Vega Ing. Freddy Chamorro

Grabar Informe Estado Informe: Borrador

Seleccionar fecha

Ingresar datos

Clic para grabar informe

Seleccionar estado del informe

Vista Crear Informe de Actividad

Visualizar Informes de Actividades en Formato PDF:

Para ver un documento en formato PDF, se debe seleccionar la opción "Consulta y edición de informes" del menú Informes de Actividades, ante este evento el sistema mostrará la siguiente vista en la cual debe establecer el rango de fechas y el tipo de actividad complementaria a consultar y después dar clic en el botón de consulta.

Registro y Control de Actividades Complementarias de Docentes

Consulta de Informes de Actividades

Fecha Inicial: 01/03/2015 Fecha Final: 07/04/2015 Actividades Asignadas: Dirección de Tesis de Grado

Fecha Registro	Actividad Complementaria	Estado	Horas Aplicadas	Total Horas	Año y Mes	Opciones
04/04/2015	Dirección de Tesis de Grado	Confirmado	14	14	2015-05	<div style="border: 1px solid green; padding: 2px;">Clic ver en pantalla</div> <div style="border: 1px solid green; padding: 2px;">Clic exportar a PDF</div>

Vista para consultar informes de actividades

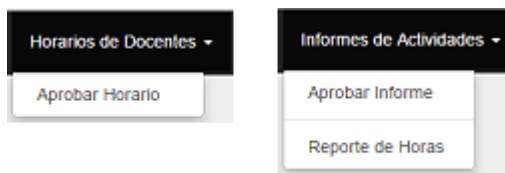
Al ejecutar el botón consultar después de haber seleccionado las fechas inicial y final y el tipo de actividad a consultar, el sistema muestra la lista de registros pertenecientes al usuario autenticado, a continuación para visualizar el informe en formato PDF en pantalla se da clic en el botón exportar y obtendremos la siguiente vista:

UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO						
INFORME DE ACTIVIDADES COMPLEMENTARIAS						
FACULTAD: Unidad de Estudios a Distancia				CARRERA: Ingeniería en Sistemas		
ACTIVIDAD: Dirección de Tesis de Grado						
DOCENTE: Cristian Gabriel Zambrano Vega				PERIODO: 2015-2016-Primer		Fecha: 06/04/2015
FECHA	HORARIO	HORAS	ACTIVIDAD REALIZADA	TAREAS ENCARGADAS	OBSERVACION	
15/05/2015	17:00-19:00	2	REVISION DE TESIS WILMER CEVALLOS	1. UNO 2. DOS	DESARROLLAR MANUAL DEL USUARIO.	
18/05/2015	13:00-15:00	2	REVISION DE TESIS MERLY OLIVO	HACER DIAGRAMA DE CLASES	CONTINUAR DOCUMENTACION	
20/05/2015	15:00-17:00	2	REVISION DE TESIS NELLY SILVA	SOLICITAR TRABAJO DE CAMPO	HACER CARTA PARA SOLICITAR AUTORIZACION TRABAJO CAMPO	
22/05/2015	17:00-19:00	2	REVISION DE TESIS CARLOS GUAMAN	INSTALAR EQUIPOS EN UED	HACER PRUEBAS DE COMUNICACION EN UED.	
25/05/2015	13:00-15:00	2	REVISION DE TESIS WILMER CEVALLOS	CREAR MANUAL USUARIO	DESARROLLAR MANUAL DEL USUARIO APLICACION RAC-UTEQ.	
27/05/2015	15:00-17:00	2	REVISION DE TESIS EDEL ROSERO	DESARROLLAR UNIDAD 1, 2 Y 3.	ENTREGAR INFORMACION DE LAS TRES UNIDADES.	
29/05/2015	17:00-19:00	2	REVISION DE TESIS WILMER CEVALLOS	ENTREGAR MANUAL.	SOLICITAR TRIBUNAL DE TESIS.	
Ing. Domingo			Ing. Cristian Zambrano Vega		Ing. Freddy Cham...	

Vista del informe de actividad complementaria en formato PDF.

Como se observa la vista también muestra los botones de página completa, acercar, alejar, grabar, que son acciones que podemos realizar con el archivo actual.

Opciones del menú del Usuario Coordinador:



Opciones del menú del usuario coordinador

Aprobar Horario:

Esta interfaz le permite al usuario coordinador de carrera, aprobar los horarios de los docentes, para este efecto primero debe seleccionar el docente de la

lista desplegable, una vez seleccionado el docente, el sistema mostrará el horario que previamente el mismo docente creó para ingresar los informes de actividades, la siguiente figura muestra el diseño de la vista, en la cual se tiene el botón Aprobar Horario, mediante el cual el coordinador de carrera aprobará el horario del docente dando clic en el mismo.

Registro y Control de Actividades Complementarias de Docentes

Aprobar Horario de Actividades

Docente: Cristian Gabriel Zambrano Vega

Estado: StandBy

Tipo de Actividad	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Dirección de Tesis de Grado	13.00-15.00						
Dirección de Tesis de Grado			15.00-17.00				
Dirección de Tesis de Grado					17.00-19.00		

Vista Aprobar horario del docente

Aprobar Informe de Actividad:

En la siguiente figura tenemos la vista inicial donde el coordinador de carrera seleccionará el período de clases, el año y mes de los informes ingresados y el tipo de actividad a consultar para que el sistema muestre todas las actividades registradas en el año y mes indicado, y que hayan sido registrados como confirmados. De esta lista el usuario autenticado como coordinador de carrera, podrá seleccionar el informe que inicialmente revisará y posteriormente aprobará.

Registro y Control de Actividades Complementarias de Docentes

Aprobación de Informes de Actividades

Período: 2015-2016 - Prime | Año-Mes: 2015-Mayo | Tipo Actividad: Dirección de Tesis de Grado

Fecha Registro	Nombre del Docente	Total Horas	Horas Aplicadas	Horas Incumplidas	Opciones
04/04/2015	Cristian Gabriel Zambrano Vega	14	14	0	[Icon]

Vista consulta de informes de actividades para aprobar

Una vez seleccionado el informe que el coordinador desea aprobar, el sistema muestra los datos de este documento, como se detalla en la siguiente figura, en la cual se observa la opción del botón Aprobar Informe, mismo que al ejecutarse actualiza el estado del informe como Aprobado para que sea compilado en los cálculos de horas cumplidas del docente.

 UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO INFORME DE ACTIVIDADES COMPLEMENTARIAS						
FACULTAD: Unidad de Estudios a Distancia				CARRERA: Ingeniería en Sistemas		
ACTIVIDAD: Dirección de Tesis de Grado						
DOCENTE: Cristian Gabriel Zambrano Vega				PERIODO: 2015-2016-Primero	Fecha: 04/04/2015	
FECHA	HORARIO	HORAS	ACTIVIDAD REALIZADA	TAREAS ENCARGADAS	OBSERVACION	
15/05/2015	17:00-19:00	2	REVISION DE TESIS WILMER CEVALLOS	1. UNO 2. DOS	DESARROLLAR MANUAL DEL USUARIO.	
18/05/2015	13:00-15:00	2	REVISION DE TESIS MERLY OLIVO	HACER DIAGRAMA DE CLASES	CONITNUAR DOCUMENTACION	
20/05/2015	15:00-17:00	2	REVISION DE TESIS NELLY SILVA	SOLICITAR TRABAJO DE CAMPO	HACER CARTA PARA SOLICITAR AUTORIZACION TRABAJO CAMPO	
22/05/2015	17:00-19:00	2	REVISION DE TESIS CARLOS GUAMAN	INSTALAR EQUIPOS EN UED	HACER PRUEBAS DE COMUNICACION EN UED.	
25/05/2015	13:00-15:00	2	REVISION DE TESIS WILMER CEVALLOS	CREAR MANUAL USUARIO	DESARROLLAR MANUAL DEL USUARIO APLICACION RAC-UTEQ.	
27/05/2015	15:00-17:00	2	REVISION DE TESIS EDEL ROSERO	DESARROLLAR UNIDAD 1, 2 Y 3.	ENTREGAR INFORMACION DE LAS TRES UNIDADES.	
27/05/2015	15:00-17:00	2	REVISION DE TESIS EDEL ROSERO	DESARROLLAR UNIDAD 1, 2 Y 3.	ENTREGAR INFORMACION DE LAS TRES UNIDADES.	
29/05/2015	17:00-19:00	2	REVISION DE TESIS WILMER CEVALLOS	ENTREGAR MANUAL	SOLICITAR TRIBUNAL DE TESIS.	

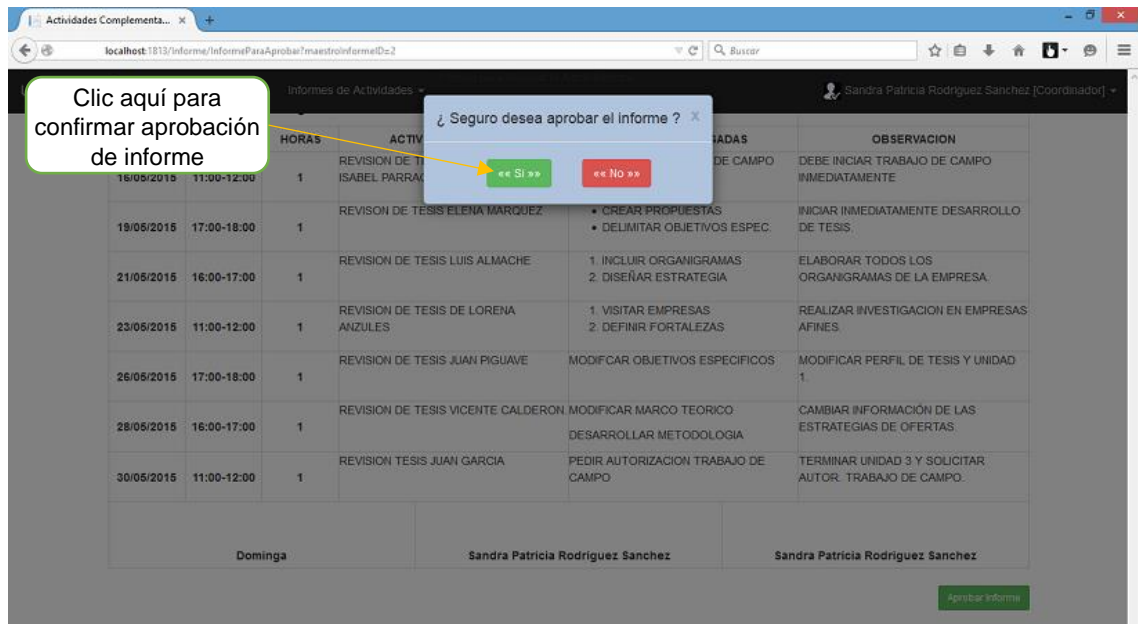
Clic aquí para aprobar informe

Ing. Dominga
Ing. Cristian Zambrano Vega
Ing. Freddy Chamorro

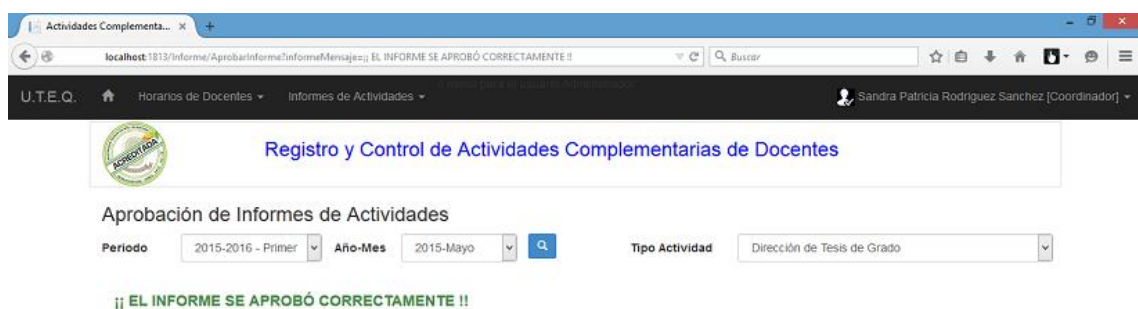
Aprobar Informe

Vista aprobar informe de actividad

Al ejecutarse el botón Aprobar Informe, el sistema mostrará la siguiente pantalla modal en la cual solicita que se confirme la aprobación. De ser positiva la respuesta en esta pregunta, el sistema procederá a registrarlo como aprobado, grabando el código del usuario que lo procesó y la fecha en la cual se actualizó, emitiendo un mensaje del correcto procesamiento, como se indica a continuación:

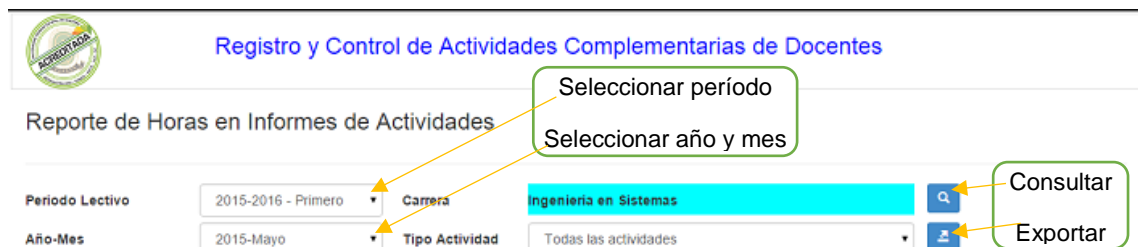


Pantalla modal de confirmación de aprobación de informe



Vista que indica la aprobación correcta del informe

Reporte de Horas Cumplidas:



Vista reporte de horas cumplidas

En esta interfaz se selecciona el período de clases, el año y mes y el tipo de actividad, para consultar los informes realizados y el número de horas

cumplidas, con la opción también de poder descargar esta información en archivos con el formato de Microsoft Excel, como se muestra en la siguiente figura:

Fecha Registro	Nombre de la Actividad	Estado	Horas al Mes	Horas Cumplidas	Año y Mes
04/04/2015	Dirección de Tesis de Grado	Confirmado	14	14	2015-05
TOTALES			14	14	

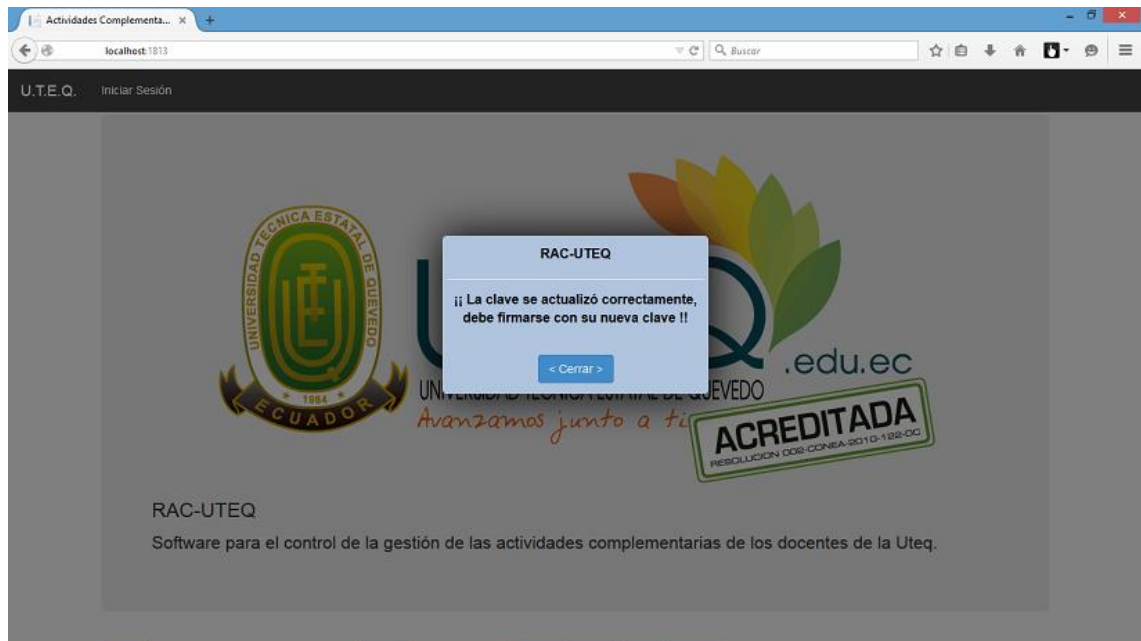
Vista reporte horas cumplidas en Microsoft Excel

Opción cambiar clave del usuario actual:

La siguiente figura muestran la interfaz para que el usuario pueda cambiar la clave, en esta vista se debe ingresar la clave anterior, la nueva clave, confirmar la nueva clave y por último dar clic en el botón Cambiar para que el sistema proceda con la actualización de los datos requeridos.

Vista cambiar clave del usuario actual

Si el usuario ejecuta el botón Cambiar, el sistema validará inmediatamente que la clave anterior sea la indicada y que la nueva clave esté confirmada, de ser correcta la información solicitada se procederá a la actualización de los campos requeridos correspondiente al registro del usuario actual, por último el sistema mostrará una pantalla modal indicando el correcto proceso de actualización de los datos, solicitando el reingreso al sistema con la nueva clave.



Pantalla modal que indica la correcta actualización de la clave del usuario

UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO
UNIDAD DE ESTUDIOS A DISTANCIA
CARRERA DE SISTEMAS



OBJETIVO:

Desarrollar la aplicación web para el control de la gestión de las actividades complementarias de los docentes de la Universidad Técnica Estatal de Quevedo.

1.- ¿Está de acuerdo con el proceso actual de informe de actividades complementarias?

- 1-SI []
- 2-NO []
- 3-DA IGUAL []

2.- ¿Qué parte del proceso de registro y control de actividades complementarias considera el de mayor problema?

- 1-REGISTRO []
- 2-COMPILACIÓN DE INFORMACIÓN []
- 3-DA IGUAL []

3.- ¿Cree usted que el uso de una aplicación web optimizaría el informe de actividades complementarias?

- 1-SI []
- 2-NO []
- 3-DA IGUAL []

4.- ¿Está de acuerdo en utilizar una aplicación web para el registro de informes de actividades complementarias?

- 1-SI []
- 2-NO []

5.- ¿Cree usted que usando una aplicación web los reportes de informes de actividades complementarias serían más rápido?

- 1-SI []
- 2-NO []

3-DA IGUAL []

6.- ¿Está usted de acuerdo en que el uso de una aplicación web reduciría el consumo de papelería?

1-SI []

2-NO []

7.- ¿Usted cree que sería una ventaja utilizar una aplicación web disponible las 24 horas del día para el registro y control de los informes de actividades complementarias?

1-SI []

2-NO []