



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
E INFORMÁTICOS**

TEMA:

**SISTEMA PARA GESTIÓN DE PROGRAMAS DE SERVICIOS
COMUNITARIOS PARA EL DIBESAU DE LA UTA.**

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniera en Sistemas Computacionales e Informáticos

SUBLÍNEA DE INVESTIGACIÓN: Aplicaciones Web.

**AUTOR: Núñez Rodríguez Amalia Elizabeth
TUTOR: Ing. Clay Fernando Aldás Flores, Mg**

Ambato - Ecuador
Mayo, 2016

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Proyecto de Investigación sobre el Tema:

“SISTEMA PARA GESTIÓN DE PROGRAMAS DE SERVICIOS COMUNITARIOS PARA EL DIBESAU DE LA UTA.”, de la señorita Amalia Elizabeth Núñez Rodríguez, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato

Ambato, Mayo de 2016

EL TUTOR

Ing. Clay Fernando Aldás Flores, Mg

AUTORÍA

El presente Proyecto de Investigación titulado: Sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad de la autora.

Ambato, Mayo de 2016

Amalia Elizabeth Núñez Rodríguez

CC:1804600748

DERECHOS DE AUTORA

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, Mayo de 2016

Amalia Elizabeth Núñez Rodríguez

CC:1804600748

APROBACIÓN COMISIÓN CALIFICADORES

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Oswaldo Paredes e Ing. Jaime Ruíz, revisó y aprobó el Informe Final del Proyecto de Investigación titulado “Sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA”, presentado por la señorita Amalia Elizabeth Núñez Rodríguez de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

Ing. Mg. Vicente Morales

PRESIDENTE DEL TRIBUNAL

Ing. Mg. Oswaldo Paredes

Ing. Mg. Ing. Jaime Ruíz

DOCENTE CALIFICADOR

DOCENTE CALIFICADOR

DEDICATORIA

El presente trabajo se lo dedico a mis padres Isaías Núñez y Elevación Rodríguez quienes me han enseñado que la perseverancia es el motor fundamental para conseguir lo que nos proponemos en la vida, a mis hermanos que de una u otra manera han estado a mi lado brindándome su ayuda y sus consejos, a mi cuñada Mirian Taboada que ha sido una hermana más y que gracias a sus consejos y apoyo he podido salir adelante.

Y a todas aquellas personas que me han alentado a lo largo de mi carrera universitaria con palabras de aliento y apoyo incondicional.

Amalia Elizabeth Núñez Rodríguez.

AGRADECIMIENTO

En primer lugar a Dios que sin él nada hubiese sido posible, luego a mis padres que gracias a su comprensión y apoyo me han permitido cumplir mis metas.

Así también al Ing. Clay Fernando Aldás Flores quien me ha sabido inculcar sus grandes conocimientos, al Ing. Oswaldo Paredes quien ha sido un gran apoyo a lo largo de mi carrera universitaria, a la Ing. Pilar Urrutia que siempre ha sabido ayudar a todo aquel que acude a ella.

Agradezco a toda mi familia y a mis compañeras y amigas Maritza, Jimena y Gabriela por alentarme a seguir adelante.

Amalia Elizabeth Núñez Rodríguez.

ÍNDICE

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN COMISIÓN CALIFICADORA	v
Dedicatoria	vi
Agradecimiento	vii
Introducción	xxvi
CAPÍTULO 1 El problema	1
1.1 Tema de Investigación	1
1.2 Planteamiento del Problema	1
1.3 Delimitación	3
1.4 Justificación	3
1.5 Objetivos	4
1.5.1 General	4
1.5.2 Específicos	4
CAPÍTULO 2 Marco Teórico	5
2.1 Antecedentes Investigativos	5
2.2 Fundamentación Teórica	6
2.2.1 Gestión de Programas	6
2.2.2 Sistema Informático	6
2.2.2.1 Partes de un Sistema Informático	6
2.2.3 La Web	7
2.2.4 Tecnologías Web	7
2.2.5 Aplicaciones Web	7
2.2.5.1 Arquitectura de las Aplicaciones Web	7

2.2.5.2	Ventajas y Desventajas de las Aplicaciones Web . . .	8
2.2.6	Servidor Web	9
2.2.7	Navegador Web	9
2.2.8	Sistema de Gestión de Bases de Datos	10
2.2.9	Base de Datos	10
2.3	Propuesta de Solución	11
CAPÍTULO 3 Metodología		12
3.1	Modalidad Básica de la Investigación	12
3.2	Recolección de Información	12
3.3	Población y Muestra	13
3.4	Procesamiento y Análisis de Datos	13
3.5	Desarrollo del Proyecto	13
CAPÍTULO 4 Desarrollo de la Propuesta		15
4.1	Recolección y Análisis de la Información	15
4.2	Definición Ciclo de Vida del Software	16
4.2.1	Modelos del Ciclo de Vida del Software	16
4.3	Definición del Modelo del Ciclo de Vida del Software para el Desarrollo del Sistema	18
4.3.1	Modelo en Cascada	18
4.3.1.1	Fases del Modelo en Cascada	18
4.4	Fase de Análisis	20
4.4.1	Análisis y Especificación de Requerimientos	20
4.4.1.1	Requerimientos Funcionales	20
4.4.1.2	Requerimientos no Funcionales	45
4.4.2	Definición de la Arquitectura de la Aplicación Web	47
4.4.2.1	Arquitectura Cliente/Servidor	48
4.4.3	Conceptualización de Herramientas	51
4.4.3.1	Sistema de Gestión de Base de Datos (SGBD)	51
4.4.3.2	Herramientas de Programación	52
4.4.4	Selección de la Tecnología de Acceso a Datos	56
4.4.4.1	ADO.NET	57
4.4.4.2	LINQ to SQL	59
4.4.4.3	Entity Framework	63
4.4.4.4	Cuadro Comparativo Tecnologías de Acceso a Datos	65
4.4.4.5	Entity Framework (EF)	67
4.5	Fase de Diseño	76

4.5.1	Diagramas de Secuencia	76
4.5.2	Diseño del Modelo de Base de Datos	89
4.5.3	Realización del Diccionario de Datos	93
4.5.4	Diseño de Interfaces	104
4.6	Fase de Implementación	112
4.6.1	Desarrollo de la Aplicación Web	112
4.7	Fase de Pruebas	117
4.7.1	Pruebas de Funcionamiento	117
4.7.1.1	Pruebas de Caja Blanca	117
4.7.1.2	Pruebas de Caja Negra	120
4.7.2	Implantación del Sistema Web	123
CAPÍTULO 5 Conclusiones y Recomendaciones		125
5.1	CONCLUSIONES	125
5.2	RECOMENDACIONES	127
Bibliografía		128
ANEXOS		135

ÍNDICE DE TABLAS

4.1	Caso de Uso: Iniciar Sesión	21
4.2	Caso de Uso: Iniciar Sesión	22
4.3	Caso de Uso: Ingresar Cargos-Responsables	22
4.4	Caso de Uso: Modificar Cargos-Responsables	23
4.5	Caso de Uso: Eliminar Cargos-Responsables	23
4.6	Caso de Uso: Eliminar Cargos-Responsables	24
4.7	Caso de Uso: Ingresar Especialidades-Responsables	24
4.8	Caso de Uso: Ingresar Especialidades-Responsables	25
4.9	Caso de Uso: Modificar Especialidades-Responsables	25
4.10	Caso de Uso: Modificar Especialidades-Responsables	26
4.11	Caso de Uso: Eliminar Especialidades-Responsables	26
4.12	Caso de Uso: Eliminar Especialidades-Responsables	27
4.13	Caso de Uso: Ingresar Representantes-Programas	27
4.14	Caso de Uso: Ingresar Representantes-Programas	28
4.15	Caso de Uso: Modificar Representantes-Programas	28
4.16	Caso de Uso: Modificar Representantes-Programas	29
4.17	Caso de Uso: Eliminar Representantes-Programas	29
4.18	Caso de Uso: Eliminar Representantes-Programas	30
4.19	Caso de Uso: Ingresar Programas	30
4.20	Caso de Uso: Ingresar Programas	31
4.21	Caso de Uso: Modificar Programas	31
4.22	Caso de Uso: Modificar Programas	32
4.23	Caso de Uso: Eliminar Programas	32
4.24	Caso de Uso: Eliminar Programas	33
4.25	Caso de Uso: Ingresar Representantes-Proyectos	33
4.26	Caso de Uso: Ingresar Representantes-Proyectos	34
4.27	Caso de Uso: Modificar Representantes-Programas	34
4.28	Caso de Uso: Modificar Representantes-Programas	35
4.29	Caso de Uso: Eliminar Representantes-Proyectos	35
4.30	Caso de Uso: Eliminar Representantes-Proyectos	36

4.31	Caso de Uso: Ingresar Proyectos	36
4.32	Caso de Uso: Ingresar Proyectos	37
4.33	Caso de Uso: Modificar Proyectos	37
4.34	Caso de Uso: Modificar Proyectos	38
4.35	Caso de Uso: Eliminar Proyectos	38
4.36	Caso de Uso: Eliminar Proyectos	39
4.37	Caso de Uso: Ingresar Asignación de responsables a Proyectos	39
4.38	Caso de Uso: Ingresar Asignación de responsables a Proyectos	40
4.39	Caso de Uso: Modificar Asignación de responsables a Proyectos	40
4.40	Caso de Uso: Modificar Asignación de responsables a Proyectos	41
4.41	Caso de Uso: Eliminar Asignación de responsables a Proyectos	41
4.42	Caso de Uso: Eliminar Asignación de responsables a Proyectos	42
4.43	Caso de Uso: Ingresar Actividades	42
4.44	Caso de Uso: Ingresar Actividades	43
4.45	Caso de Uso: Modificar Actividades	43
4.46	Caso de Uso: Modificar Actividades	44
4.47	Caso de Uso: Eliminar Actividades	44
4.48	Caso de Uso: Eliminar Actividades	45
4.49	Cuadro Comparativo Sistemas de Gestión de Base de Datos.	51
4.50	Cuadro Comparativo Herramientas de Programación.	53
4.51	Cuadro Comparativo Tecnologías de Acceso a Datos	65
4.52	Cuadro Comparativo Tecnologías de Acceso a Datos	66
4.53	Cuadro de los Operadores de Consulta para LINQ To Entities (L2E) y Entity SQL (eSQL).	74
4.54	Cuadro de los Operadores de Consulta para LINQ To Entities (L2E) y Entity SQL (eSQL).	75
4.55	Diccionario de Datos Tabla Cargos	93
4.56	Diccionario de Datos Tabla Cargos	93
4.57	Diccionario de Datos Tabla Especialidades	93
4.58	Diccionario de Datos Tabla Especialidades	94
4.59	Diccionario de Datos Tabla Responsables Programa	94
4.60	Diccionario de Datos Tabla Responsables Programa	94
4.61	Diccionario de Datos Tabla Responsables Programa	95
4.62	Diccionario de Datos Tabla Títulos Responsable Programa	95
4.63	Diccionario de Datos Tabla Títulos Responsable Programa	95
4.64	Diccionario de Datos Tabla Programas Generales	96
4.65	Diccionario de Datos Tabla Programas Generales	96

4.66	Diccionario de Datos Tabla Responsables Proyecto	96
4.67	Diccionario de Datos Tabla Responsables Proyecto	97
4.68	Diccionario de Datos Tabla Responsables Proyecto	97
4.69	Diccionario de Datos Tabla Títulos Responsable Proyecto	98
4.70	Diccionario de Datos Tabla Títulos Responsable Proyecto	98
4.71	Diccionario de Datos Tabla Proyectos	98
4.72	Diccionario de Datos Tabla Proyectos	99
4.73	Diccionario de Datos Tabla Proyectos Responsables	99
4.74	Diccionario de Datos Tabla Proyectos Responsables	99
4.75	Diccionario de Datos Tabla Lugares Actividades	100
4.76	Diccionario de Datos Tabla Lugares Actividades	100
4.77	Diccionario de Datos Tabla Actividades Proyectos	100
4.78	Diccionario de Datos Tabla Actividades Proyectos	101
4.79	Diccionario de Datos Tabla Horarios	101
4.80	Diccionario de Datos Tabla Horarios	102
4.81	Diccionario de Datos Tabla Horarios	102
4.82	Diccionario de Datos Tabla Horarios	103
4.83	Diccionario de Datos Tabla Problemas	103
4.84	Diccionario de Datos Tabla Problemas	103
4.85	Diccionario de Datos Tabla Soluciones	103
4.86	Diccionario de Datos Tabla Soluciones	104
4.87	Diccionario de Datos Tabla Soluciones	104
4.88	Cuadro de Pruebas de Sesión Administrativo	118
4.89	Cuadro de Pruebas de Sesión Administrativo	119
4.90	Cuadro de Pruebas de Sesión Administrativo	123
4.91	Cuadro de Pruebas de Sesión Administrativo	124
B.1	Guía de Observación	183

ÍNDICE DE FIGURAS

2.1	Arquitectura Cliente/Servidor.	8
4.1	Modelo de ciclo de vida en cascada	19
4.2	Diagrama de Caso de Uso de los Procesos de Gestión de Servicios Comunitarios	47
4.3	Arquitectura Cliente/Servidor.	48
4.4	Arquitectura de ADO.NET.	58
4.5	Arquitectura y componentes Entity Framework.	64
4.6	Arquitectura y componentes Entity Framework.	68
4.7	Secciones de metadatos de un modelo EDMX.	71
4.8	Diagrama de Secuencia Iniciar sesión	77
4.9	Diagrama de Secuencia Ingresar Cargos-Responsables	77
4.10	Diagrama de Secuencia Actualizar Cargos-Responsables	78
4.11	Diagrama de Secuencia Eliminar Cargos-Responsables	78
4.12	Diagrama de Secuencia Ingresar Especialidades-Responsables	79
4.13	Diagrama de Secuencia Actualizar Especialidades-Responsables	79
4.14	Diagrama de Secuencia Eliminar Especialidades-Responsables	80
4.15	Diagrama de Secuencia Ingresar Responsables-Programas	80
4.16	Diagrama de Secuencia Actualizar Responsables-Programas	81
4.17	Diagrama de Secuencia Eliminar Responsables-Programas	81
4.18	Diagrama de Secuencia Ingresar Programas	82
4.19	Diagrama de Secuencia Actualizar Programas	82
4.20	Diagrama de Secuencia Eliminar Programas	83
4.21	Diagrama de Secuencia Ingresar Responsables-Proyectos	83
4.22	Diagrama de Secuencia Actualizar Responsables-Proyectos	84
4.23	Diagrama de Secuencia Eliminar Responsables-Proyectos	84
4.24	Diagrama de Secuencia Ingresar Proyectos	85
4.25	Diagrama de Secuencia Actualizar Proyectos	85
4.26	Diagrama de Secuencia Eliminar Proyectos	86
4.27	Diagrama de Secuencia Ingresar Asignación de responsables a Proyectos	86

4.28 Diagrama de Secuencia Actualizar Asignación de responsables a Proyectos	87
4.29 Diagrama de Secuencia Eliminar Asignación de responsables a Proyectos	87
4.30 Diagrama de Secuencia Ingresar Actividades	88
4.31 Diagrama de Secuencia Actualizar Actividades	88
4.32 Diagrama de Secuencia Eliminar Actividades	89
4.33 Entity Data Model (EDM) dbe_DUBESAU	90
4.34 Entity Data Model (EDM) dbe_DUBESAU	91
4.35 Entity Data Model (EDM) dbe_DUBESAU	92
4.36 Diseño Interfaz Página Inicio	105
4.37 Diseño Interfaz Inicio de Sección	106
4.38 Diseño Pagina Principal Menú Administrador	106
4.39 Diseño Interfaz Submenús	107
4.40 Diseño Interfaz Registros	107
4.41 Diseño Interfaz Ingreso Nuevos Registros	108
4.42 Diseño Interfaz Seleccionar un Registro	108
4.43 Diseño Interfaz Mostrar Datos	109
4.44 Diseño Interfaz Buscar Datos	109
4.45 Diseño Interfaz Seleccionar un Registro	109
4.46 Diseño Interfaz Calendario	110
4.47 Diseño Interfaz Modificar Registros	110
4.48 Diseño Interfaz Eliminar Registros	111
4.49 Diseño Interfaz Ver Reporte	111
4.50 Diseño Interfaz de Página Grid Anidados	112
4.51 Pruebas de Sesión Administrativo	118
4.52 Pruebas de Sesión Administrativo	119
4.53 Pruebas Iniciar Sección	120
4.54 Pruebas Iniciar Sección	121
4.55 Pruebas Ingreso Programas	121
4.56 Pruebas Ingreso Actividades	122
4.57 Pruebas Ingreso Horarios	122
A.1 Menú Administrador	136
A.2 Pantalla de Inicio del Sistema	137
A.3 Pantalla de Ingreso al Sistema	138
A.4 Menú Administrador	138
A.5 Pantalla Menú Responsables/Datos Laborales	140
A.6 Pantalla de Cargos	140

A.7	Pantalla de Ingreso de un Nuevo Registro de Cargo	141
A.8	Proceso de Modificar un Registro de Cargos	141
A.9	Pantalla de Modificar un Registro de Cargos	142
A.10	Pantalla de Eliminación del Registro de Cargos	142
A.11	Pantalla Registro de Especialidades	143
A.12	Pntalla Menú de Programas	143
A.13	Opción Responsables/Datos Personales	144
A.14	Pantalla Responsables/Datos Personales	144
A.15	Pantalla de Nuevo Registro Responsables/Datos Personales	144
A.16	Pantalla Seleccionar el Cargo	145
A.17	Mostrar Datos de Cargos	145
A.18	Buscar Datos de Cargos	145
A.19	Seleccionar el Cargo	146
A.20	Error Cédula Incorrecta	146
A.21	Error Campos Requeridos	147
A.22	Error Correo Electrónico	147
A.23	Datos Guardados Responsables Programas Datos Personales	148
A.24	Pantalla de Modificación Responsables Programas Datos Personales	148
A.25	Pantalla de Eliminación Responsables Programas Datos Personales	148
A.26	Ver Reporte y Exportar a Pdf	149
A.27	Ver Reporte en Pdf	149
A.28	Opción Responsables/Datos Académicos	150
A.29	Pantalla Responsables/Datos Académicos	150
A.30	Pantalla de Ingreso Nuevo Dato Académico	150
A.31	Seleccionar el Responsable del Programa al que Corresponde el Título	151
A.32	Ingresar la información y Guardar	151
A.33	Información Guardada Correctamente	151
A.34	Pantalla de Modificar Responsables/Datos Académicos	152
A.35	Pantalla de Eliminar Responsables/Datos Académicos	152
A.36	Ver Reporte y Exportar a Pdf	153
A.37	Opción Nuevos Programas	153
A.38	Pantalla de Nuevos Programas	153
A.39	Pantalla Ingreso Nuevos Programas	154
A.40	Seleccionar Responsable del Programa	154
A.41	Seleccionamos la Fecha de Inicio del Programa	155
A.42	Seleccionamos la Fecha de Fin del Programa	155
A.43	Error La Fecha de Inicio no Puede ser Menor o Igual a la fecha Fin	156

A.44	Mostrar Datos Guardados	156
A.45	Pantalla Modificar Programas	157
A.46	Pantalla Eliminar Programas	157
A.47	Ver Informe	158
A.48	Pantalla Menú de Proyectos	158
A.49	Pantalla Proyectos Responsables/Datos Personales	159
A.50	Pantalla Proyectos Responsables/Datos Académicos	159
A.51	Opción Nuevos Proyectos	160
A.52	Pantalla Nuevos Proyectos	160
A.53	Pantalla Ingreso Nuevos Proyectos	160
A.54	Seleccionar el Programa al que Corresponde el Proyecto.	161
A.55	Error Fechas	161
A.56	Error Campos Obligatorios	162
A.57	Ver Informe	162
A.58	Pantalla Responsable Proyecto	163
A.59	Pantalla Asignar Responsable Proyecto	163
A.60	Seleccionar el Responsable del Proyecto	164
A.61	Ingresar la información y Guardar	164
A.62	Información Guardada Correctamente	164
A.63	Pantalla Eliminar Responsable/Proyecto	165
A.64	Pantalla Menú Actividades	165
A.65	Opción Actividades Lugares	166
A.66	Pantalla Lugares Actividades	166
A.67	Pantalla de Ingreso Nuevo Registro de Lugares	167
A.68	Pantalla de Modificar un Registro de Lugares	167
A.69	Pantalla de Eliminación del Registro de lugares	168
A.70	Ver Informe	168
A.71	Opción Actividades Lugares	169
A.72	Pantalla Actividades Proyectos	169
A.73	Pantalla de Ingreso Nuevo Registro de Actividades Proyecto	170
A.74	Seleccionar el Proyecto	170
A.75	Seleccionar el Lugar	170
A.76	Información Guardada Correctamente	171
A.77	Pantalla Horarios	171
A.78	Pantalla de Ingreso Nuevo Horario Actividad	172
A.79	Seleccionar el Proyecto	172
A.80	Seleccionar el Lugar	173

A.81 Información Guardada Correctamente	173
A.82 Pantalla Modificar Horario Actividad	174
A.83 Pantalla Eliminar Horario Actividad	174
A.84 Pantalla Opción Indicadores	175
A.85 Pantalla de Ingreso Indicadores	175
A.86 Pantalla de Ingreso Indicadores	175
A.87 Pantalla Calculo de Indicadores	176
A.88 Pantalla Mostrar Actividad	176
A.89 Pantalla Seleccionar Actividad	177
A.90 Pantalla Mostrar Indicadores	177
A.91 Pantalla Problemas Soluciones	177
A.92 Pantalla Mostrar Actividades Proyecto	178
A.93 Pantalla Ingreso Nuevo Problema	178
A.94 Pantalla Seleccionar Problema	179
A.95 Pantalla Ingreso Nuevo Solución	179
A.96 Pantalla Información Ingresada Correctamente	180
A.97 Pantalla Menú Informes	180
A.98 Pantalla Menú Informes	181
A.99 Pantalla Menú Informes	181

Resumen

La Universidad Técnica de Ambato (UTA) cuenta con varios departamentos, uno de ellos es el DIBESAU (Dirección de Bienestar Estudiantil y Asistencia Universitaria) el cual brinda diferentes tipos de servicios a la comunidad universitaria, uno de los más importantes son los programas de servicios comunitarios; tales como, programas de servicios para la salud, programas de educación para la salud y programas de recreación universitaria. Cada programa cuenta con varios proyectos en los cuales se realizan distintas actividades para la comunidad.

Los programas de servicios comunitarios se llevan de forma manual, por este motivo y al evidenciar la necesidad de mejorar y agilizar el proceso de gestión de programas de servicios comunitarios para el DIBESAU de la UTA, esta investigación propone el desarrollo del sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA, para facilitar la manipulación de la información, acceso a los datos, consultas oportunas, a la vez permitir la confidencialidad, integridad, disponibilidad y seguridad de la información, ayudando en la gestión de programas de servicios comunitarios, mediante la automatización del mismo.

Para el desarrollo de este proyecto, se realizó el levantamiento de requerimientos mediante entrevistas con el personal del DIBESAU y observación de la gestión de programas de servicios comunitarios.

La etapa del desarrollo de la propuesta sigue el modelo en cascada del ciclo de vida del software, el cual es quizás el más ampliamente utilizado en el desarrollo de software, sus fases son: análisis, diseño, implementación, pruebas y mantenimiento, a través de las cuales se realizó el proceso de desarrollo del software.

Con la implantación del Sistema Web se permite la administración de los siguientes procesos: registro de los programas, responsables, proyectos, actividades e indicadores; así también la asignación de horarios a las actividades, informes de cada uno de los procesos, informes generales en los que se muestra toda la información relevante de la gestión de programas comunitarios, para el personal administrativo se

nota una disminución en la documentación física y en la pérdida de tiempo; además se obtiene rapidez y eficiencia al momento de la recuperación de la información, ayudando así a la toma de decisiones oportunas.

Abstract

Ambato Technical University (UTA) has several departments, one of them is the DIBESAU (Directorate of Student Welfare and Assistance University), the which provides various services to the university community, one of the most important are the community services programs; such as, programs of health services, programs of health education and recreation programs university. The programs have several projects in which various community activities are performed.

Community services programs are carried out manually, for this reason and to also the need to improve and streamline processes of managing community service programs for the DIBESAU of the UTA, this research proposes the development of the management system for community services programs for the DIBESAU of the UTA, to facilitate the manipulation of information, access to data, consultations, while allowing the confidentiality, integrity, availability and security of information, this help in the management of community service programs by automating.

For the development of this project, it was performed requirements gathering was conducted through interviews with staff of the DIBESAU and the observation of management of community services programs.

The stage of development of the proposal follows the waterfall model of software lifecycle, it which is perhaps the most widely used in software development, its phases are, analysis phase, design phase, implementation phase, phase of testing and phase maintenance, through of these the software development process was performed.

With the implementation of the Web system, the administration of the registration processes, programs, managers, projects, activities and beneficiaries, also it allow the allocation of schedules on the activities, also it produces reports on each of the processes, also general reports that all relevant information for the management of Community programs, for administrative staff shows a noticeable decrease in the physical documentation, loss of time; also it obtained speed and efficiency at the

time of information recovery, this helping to make timely decisions.

Glosario de términos

Abstracción

Mantener separadas las diferentes partes de un sistema.

XML

(eXtensible Markup Language). Es un Lenguaje de Etiquetado Extensible muy simple, pero estricto, que juega un papel fundamental en el intercambio de una gran variedad de datos. Su función principal es describir datos, XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Herencia

Es una propiedad que permite que los objetos sean creados a partir de otros ya existentes, obteniendo características (métodos y atributos) similares a los ya existentes. Es la relación entre una clase general y otra clase mas específica.

API

(Application Programming Interface - Interfaz de Programación de Aplicaciones). Grupo de rutinas (conformando una interfaz) que provee un sistema operativo, una aplicación o una biblioteca, que definen cómo invocar desde un programa un servicio que éstos prestan.

LINQ

(Language INtegrated Query) no es más que un modo de expresar consultas similares al sql normal.

Encapsulamiento

La encapsulación es un mecanismo que consiste en organizar datos y métodos de una estructura, conciliando el modo en que el objeto se implementa, es decir, evitando el acceso a datos por cualquier otro medio distinto a los especificados.

ORM

Object-Relational Mapping, o lo que es lo mismo, mapeo de objeto-relacional, es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador.

Expresión Lambda

Una expresión lambda es una función anónima que se puede usar para crear tipos delegados o de árbol de expresión. Utilizando expresiones lambda, puede escribir

funciones locales que se pueden enviar como argumentos o devolverse como valor de llamadas a funciones. Las expresiones lambda son especialmente útiles para escribir expresiones de consulta LINQ.

Tipos Anónimos

Los Tipos Anónimos o Anonymous Types permiten crear implícitamente campos con nombre desde el código que lo requiere. Esta es una importante característica para LINQ. Como los tipos anónimos no tiene un tipo de dato con nombre, deben ser almacenados en variables declaradas usando la palabra reservada var, que le indica al compilador que use inferencia de tipos para la variable.

Intellisense

Características de las aplicaciones de Microsoft que ayudan al usuario por la toma de decisiones de forma automática. Mediante el análisis de los patrones de actividad, el software puede derivar el siguiente paso sin que el usuario tenga que declarar explícitamente. Corrección de errata automática y atajos que sugieren también caen bajo el paraguas de IntelliSense.

Entidad

Es una parte del dominio de una aplicación que se debe representar mediante datos, las entidades son las especificaciones formales de todos los detalles de la estructura de datos.

Relación

Es la conexión lógica entre entidades: por ejemplo, la conexión lógica entre el pedido de una mercancía y el cliente que lo realiza. Dado que un cliente puede tener asociados muchos pedidos, la relación entre un cliente y sus pedidos es de uno a varios.

Dominio

Se define dominio como un conjunto de valores que puede tomar un determinado atributo dentro de una entidad.

Metadatos

Los metadatos son simplemente datos sobre datos, es decir, información estructurada que describe a otra información y que permite encontrarla, gestionarla, controlarla, entenderla y preservarla en el tiempo.

EDM

Es el denominado Entity Data Model (EDM) es un modelo de entidad relación. El EDM define los datos en un formato neutro que no está restringido por la estructura

de los lenguajes de programación o las bases de datos relacionales. Los esquemas EDM se usan para especificar los detalles de las entidades y las relaciones y para implementarlos como estructuras de datos.

CSDL

Lenguaje de definición de esquemas conceptual, lenguaje basado en XML que se utiliza para definir tipos de entidades, asociaciones, contenedores de entidades, conjuntos de entidades y conjuntos de asociaciones de un modelo conceptual.

SSDL

Lenguaje de definición de esquemas de almacenamiento, Lenguaje basado en XML que se utiliza para definir los tipos de entidad, asociaciones, contenedores de entidad, conjuntos de entidades y conjuntos de asociaciones de un modelo de almacenamiento que, con frecuencia, corresponden a un esquema de base de datos.

Object Services

Object Services es un componente de Entity Framework que permite consultar, insertar, actualizar y eliminar datos. Object Services admite tanto consultas LINQ (Language-Integrated Query) como consultas Entity SQL con los tipos que se definen en un modelo Entity Data Model (EDM). Object Services materializa los datos devueltos como objetos y propaga los cambios de los objetos de vuelta al origen de datos.

Entity SQL

Entity SQL es un lenguaje de consulta independiente del almacenamiento que se parece a SQL. Entity SQL permite consultar los datos de la entidad.

LINQ to Entities

LINQ to Entities proporciona la capacidad de realizar consultas integradas en lenguajes (LINQ) que permite a los desarrolladores de software escribir consultas contra el modelo conceptual de Entity Framework.

INTRODUCCIÓN

El presente Proyecto de investigación denominado: “SISTEMA PARA GESTIÓN DE PROGRAMAS DE SERVICIOS COMUNITARIOS PARA EL DIBESAU DE LA UTA.”, se lo ha dividido en los siguientes capítulos:

CAPÍTULO I, “EL PROBLEMA”, identifica el problema a investigar y además se plantea la justificación por la cual se investiga así como los objetivos a obtener los que guiarán la realización del proyecto.

CAPÍTULO II, “MARCO TEÓRICO”, muestra las investigaciones previas que sirven de soporte para el desarrollo de la investigación, además la información de estudios similares anteriormente realizados, así como los aspectos conceptuales que sustentan el tema en general, y el conjunto de conceptos y fundamentos teóricos que han sido analizados en base al problema establecido.

CAPÍTULO III, “METODOLOGÍA”, describe la metodología de investigación a utilizar y el proceso de recolección, procesamiento y análisis de la información recolectada. Además especifica de manera breve cada una de las etapas para el desarrollo del proyecto.

CAPÍTULO IV, “DESARROLLO DE LA PROPUESTA”, detalla todo el desarrollo de la propuesta de solución siguiendo el ciclo de vida del software escogido, definiendo los requisitos necesarios obtenidos del levantamiento de requisitos y documentando el proceso mediante diagramas de casos de uso, diagramas de secuencia, el diseño de la base de datos, el diccionario de datos describiendo cada una de las tablas que constan en la base de datos, el diseño de la interfaz gráfica y la implementación.

CAPÍTULO V, “CONCLUSIONES Y RECOMENDACIONES”, se establecen las conclusiones a las que se ha llegado luego del desarrollo del proyecto así como recomendaciones que el investigador ha considerado pertinentes.

CAPÍTULO 1

El problema

1.1. Tema de Investigación

Sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.

1.2. Planteamiento del Problema

A lo largo de la historia, el avance de la tecnología, la informática y el manejo de la información ha venido evolucionando favorablemente día a día, por lo cual la necesidad de contar con sistemas automatizados para el control de la información es indispensable.

Imaginar hoy en día un mundo sin el uso de sistemas informáticos es algo difícil, la automatización para la administración de la información cada vez se convierte en una necesidad entre los seres humanos. La optimización de tiempo, eficiencia en el desempeño de tareas, capacidad de almacenamiento, la obtención de información de forma rápida y flexible, dan paso para que el personal dedique un tiempo extra a buscar datos, permite que puedan dedicar ese tiempo en debatir o desarrollar iniciativas de acuerdo con esa información que ya proporciona el sistema.

El Ministerio de Telecomunicaciones y de la Sociedad de la Información y la Subsecretaría de Fomento de la Sociedad de la Información y Gobierno en Línea de Ecuador en enero, 2014 establecieron la importancia del desarrollo y utilización de las Tecnologías de la Información y Comunicaciones ya que aporta a la transformación de la matriz productiva del país a través de la transferencia y difusión de nuevas tecnologías, la generación de empleos calificados y la exportación de servicios. Ésta impacta de forma transversal y tiene efectos positivos sobre los demás sectores de la economía, induce aumentos de productividad empresarial y contribuye a diversificar

la oferta exportadora, constituyéndose en el motor más importante del crecimiento económico del siglo XXI, contribuyendo con la reducción de la pobreza y las brechas sociales.[1]

La Encuesta Nacional de Empleo Desempleo y Subempleo del INEC muestra que en el 2013, el 43,6 % de las personas de Ecuador utilizaron computadora, 6,1 puntos más que en el 2010. En el área rural el incremento es de 8,4 puntos más que en el 2010. El 32,0 % de las personas usó Internet como fuente de información, mientras el 31,7 % lo utilizó como medio de educación y aprendizaje. [2]

En la actualidad la Universidad Técnica de Ambato en adelante denominada UTA cuenta con varios departamentos uno de ellos es la Dirección de Bienestar Estudiantil y Asistencia Universitaria de aquí en adelante denominado DIBESAU, este no cuenta con un sistema para gestión de programas de servicios comunitarios; como son los programas de servicios para la salud, programas de educación para la salud y programas de recreación universitaria. Cada programa cuenta con varios proyectos, en los cuales se realizan distintas actividades para la comunidad.

Los programas de servicios comunitarios se llevan de forma manual, por lo cual se emplea demasiado tiempo, así también la información se duplica. El registro de las actividades para los proyectos se realizan manualmente, esto ocasiona el retraso de la ejecución de las actividades. La información de los responsables de los proyectos se la lleva manualmente, por ende el tiempo utilizado para realizar consultas referentes a responsables y proyectos es considerable.

El registro de los horarios de las actividades es de forma manual, ocasionando que el registro y la búsqueda de los horarios sea dificultosa y se emplee mucho tiempo para realizarla.

Los informes de los proyectos se desarrollan manualmente, provocando que estos no sean oportunos para una buena toma de decisiones.

El número de beneficiarios se calculan manualmente, por lo que se consume mucho tiempo en los cálculos. Toda la información se guarda en archivadores, exponiendo a la información a varios riesgos como robos, desastres naturales, incendios y otros; de la misma manera dificulta el acceso a la información ya que al llevarla en papel esta se acumula y el tiempo de búsqueda es extenso.

Debido al proceso manual de la información para la gestión de los programas comunitarios existe mucho retraso en el registro de actividades, beneficiarios, representantes, elaboración de informes y en la ejecución misma de los programas.

1.3. Delimitación

Área Académica: Software.

Línea de Investigación: Desarrollo de software.

Sublínea de Investigación: Aplicaciones Web.

Delimitación Espacial: El presente proyecto se desarrollará en la Universidad Técnica de Ambato (UTA), Dirección de Bienestar Estudiantil y Asistencia Universitaria (DIBESAU).

Delimitación Temporal: El desarrollo de este trabajo durará 6 meses a partir de la fecha de aprobación por parte de Consejo Directivo.

1.4. Justificación

Se procederá a desarrollar el presente proyecto, en vista de la inexistencia de un software informático para gestión de programas de servicios comunitarios para el DIBESAU de la UTA, debido a que la accesibilidad a la información es complicada, la búsqueda en archivadores requiere mucho tiempo y la información no es oportuna para la toma de decisiones.

Al evidenciar la necesidad de mejorar y automatizar la gestión de la información se ve la importancia del desarrollo de un sistema para gestión de programas de servicios comunitarios; de esta manera se facilitará el manejo de la información de los programas, el ahorro de tiempo en los registros de las actividades, representantes y beneficiarios; los cálculos de los beneficiarios por género, etnia, edad y otros serán automáticos, permitiendo que el consumo de tiempo en estos procesos sea el mínimo, de la misma manera el sistema podrá mostrar los reportes requeridos, obteniendo la información de forma rápida y precisa por lo que la toma de decisiones serán oportunas.

De la misma manera se desea brindar una mayor confiabilidad para el tratamiento de la información con su correcta manipulación, sin temor de duplicidad, pérdida de información e inaccesibilidad a los datos.

El presente trabajo es factible ya que se cuenta con toda la apertura a la información requerida, con el apoyo del personal encargado del DIBESAU y con el asesoramiento de personal especializado.

Debido a las grandes ventajas de una aplicación Web como son, la portabilidad, el ahorro en costes de hardware y software, compatibilidad, accesibilidad, actualizaciones rápidas y seguras, escalabilidad y ahorro de espacio en disco, se ha optado

por el desarrollo de una aplicación Web para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.

Los beneficiarios del presente trabajo serán los directivos del DIBESAU, los empleados, y los usuarios quienes usaran el sistema.

1.5. Objetivos

1.5.1. General

- Implantar un sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.

1.5.2. Específicos

- Analizar los procesos para la gestión de programas de servicios comunitarios del DIBESAU de la UTA.
- Analizar los requerimientos para el desarrollo del sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.
- Determinar la tecnología de acceso a datos para el desarrollo del sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.
- Diseñar el sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.
- Desarrollar el sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.
- Realizar la fase de pruebas de funcionamiento del sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.

CAPÍTULO 2

Marco Teórico

2.1. Antecedentes Investigativos

Porras Veloz Daniel Mauricio, Diseñó, desarrolló e implementó un sistema de control y asistencia del personal administrativo y de servicio para la Dirección de Talento Humano de la Universidad Central del Ecuador, el cual está enfocado en automatizar los procesos de control y asistencia , permitiendo que los actores involucrados, tengan una herramienta fácil de utilizar y a su vez que sea potente, apegada a estándares de calidad de software, parametrizable a cambios que se presenten en el futuro y principalmente segura, puesto que la aplicación gestiona datos que son de vital importancia para la institución.[3]

Chalán Oviedo Diana Margoth, desarrolló un sistema de control y adquisiciones de la Universidad Central del Ecuador, implantó un sistema que mecaniza el flujo de los procesos que se realizan, con el fin de facilitar y garantizar el correcto funcionamiento de acuerdo a los tiempos establecidos y usuarios asignados, llevando un mejor control de flujo de trabajo. Este sistema muestra de manera atractiva, rápida y eficiente la información al usuario.[4]

Pachacama Cruz Elsa Janneth, desarrolló un sistema Web, para la administración académica y capacitación virtual en el Instituto Niño Jesús de Praga (INJP), este se centra en el manejo de la información que genera el estudiante matriculado siendo el principal actor que interviene en el sistema.[5]

Sevilla Alvarado Alex Javier, desarrolló e implementó una aplicación Web para la administración de los proyectos de investigación que realiza el CENI de la Universidad Técnica de Ambato. La página Web cuenta con una interfaz intuitiva para la demostración de resultados así como para el ingreso de los diferentes usuarios y datos, facilitando su acceso a los administradores del sistema. Además con las debidas seguridades de acceso de los diferentes tipos de datos.[6]

Se tomará como antecedentes investigativos estos trabajos, que ayudarán a elaborar y tener un estudio bibliográfico que servirá como referencia para el desarrollo del sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.

2.2. Fundamentación Teórica

2.2.1. Gestión de Programas

La gestión de programas es un marco amplio que incluye la gestión de proyectos como medio para llevar a cabo los planes y objetivos establecidos. La gestión de programas se basa en una jerarquía de tres niveles.

El nivel inferior de la jerarquía se compone de gestores de proyectos asignados para hacer frente a diversos proyectos relacionados con el programa. La jerarquía media está formada por los directores de programas que supervisan el trabajo de los directores de proyectos. Los directores de programas también establecen y revisan los objetivos, coordinan las actividades a través de diferentes proyectos, y garantizan que la labor realizada alcanza las estrategias establecidas por los gerentes del nivel superior de la jerarquía. Administrativamente, proporcionan e interpretan las políticas con el fin de crear un impulso sostenible para todo el programa y examinan el progreso y resultados para asegurar que los objetivos alcanzados cumplen con la visión estratégica general del programa.[7]

2.2.2. Sistema Informático

Un sistema informático puede ser definido como un sistema de información que basa la parte fundamental de su procesamiento, en el empleo de la computación, como cualquier sistema, es un conjunto de funciones interrelacionadas, hardware, software y de recurso humano. [8]

2.2.2.1. Partes de un Sistema Informático

- Hardware: Es el conjunto de elementos físicamente visualizables en un sistema de procesamiento electrónico de datos, es el equipo propiamente dicho.
- Software: Se trata de las aplicaciones y los datos que explota el hardware.
- Personal: Está compuesto tanto por los usuarios que interactúan con los equipos hardware, como por aquellos que desarrollan y mantienen el software.

- Información descriptiva: Es el conjunto de manuales técnicos o de usuario, formularios, documentación de procedimientos o cualquier soporte que dé instrucciones, sobre el uso del sistema informático.[9]

2.2.3. La Web

El servicio World Wide Web (La Telaraña Mundial), también conocido como WWW o simplemente Web, es un Sistema de Información distribuido por Internet basado en la tecnología hipertexto/hipermedia, que proporciona una interfaz común a los distintos formatos de datos (texto, gráficos, video, audio, etc) y a los servicios de Internet existentes. [10]

2.2.4. Tecnologías Web

Las tecnologías Web sirven para acceder a los recursos de conocimiento disponibles en Internet o en las intranets utilizando un navegador. Están muy extendidas por muchas razones: facilitan el desarrollo de sistemas, su flexibilidad en términos de escalabilidad, es decir, a la hora de expandir el sistema; su sencillez de uso y que imitan la forma de relacionarse de las personas, al poner a disposición de todos el conocimiento de los demás, por encima de jerarquías, barreras formales u otras cuestiones.[11]

2.2.5. Aplicaciones Web

Son aquellas en que los usuarios acceden a ellas en un servidor Web a través de Internet o de una intranet. La popularidad de las Aplicaciones Web se debe a que basta con tener solamente una PC, un navegador Web y acceso a la Internet. La facilidad de uso, la habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software es otra razón de su enorme popularidad. [12]

Lo que distingue una aplicación Web de un mero sitio Web reside en la posibilidad que ofrece al usuario de actuar sobre la lógica de negocio en el servidor. Por lógica de negocio se entiende un conjunto de procesos que implementan las reglas de funcionamiento de la aplicación Web. [13]

2.2.5.1. Arquitectura de las Aplicaciones Web

La arquitectura de una aplicación Web es similar a la de un sitio Web, se basa en el modelo Cliente/Servidor. Como en el caso del sitio Web, se tiene el navegador en la

parte cliente, el servidor Web en la parte del servidor y una conexión de red. Pero en las aplicaciones Web hay que considerar que existe una lógica de negocio sensible a las interacciones del usuario.[13]

Existen diversas variantes de la arquitectura básica según cómo se implementen las diferentes funcionalidades de la parte servidor.

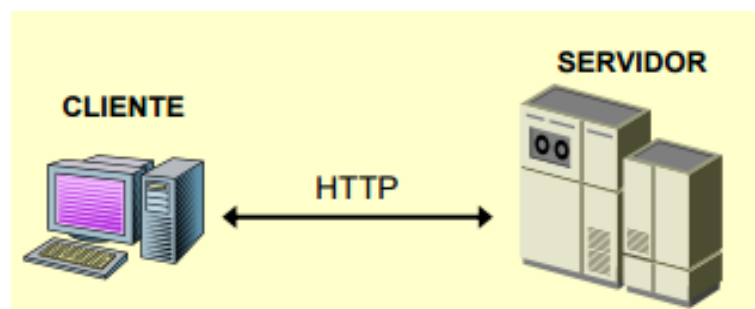


Figura 2.1: Arquitectura Cliente/Servidor.

Fuente: “DLSI – Universidad de Alicante”, <http://rua.ua.es/dspace/bitstream/10045/4412/5/03c-AplicacionesWeb.pdf>

2.2.5.2. Ventajas y Desventajas de las Aplicaciones Web

■ Ventajas

- Ahorro de tiempo, se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad, basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio en el disco duro. Actualizaciones inmediatas.
- Consumo de recursos bajo, dado que toda (o gran parte) de la aplicación no se encuentra en el ordenador, muchas de las tareas que realiza el software no consumen recursos porque se realizan desde otro ordenador.
- Multiplataforma, se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portables, es independiente del ordenador donde se utilice porque se accede a través de una página Web (sólo es necesario disponer de acceso a Internet).

- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
- Colaboración, gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartir datos por parte de varios usuarios. Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones Web.[14]

■ Desventajas

- Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo. Pero cada vez los navegadores están más preparados para mejorar en este aspecto.
- La disponibilidad depende de un tercero, el proveedor de la conexión a internet o el que provee el enlace entre el servidor de la aplicación y el cliente. [14]

2.2.6. Servidor Web

Un servidor Web es el encargado de recibir las peticiones referidas a páginas o elementos de la Web a través del protocolo http o https y de devolver el resultado de la petición, que suele ser un recurso alojado en el servidor. Normalmente es el navegador el que pide al servidor Web el recurso que desea el usuario, para finalmente recibir dicho recurso (si fue válida la petición) y traducirle si es necesario a su forma legible por el usuario (es decir la traducción de HTML la hace el navegador). [15]

2.2.7. Navegador Web

El navegador Web o navegador de Internet es el instrumento que permite a los usuarios de Internet navegar entre las distintas páginas de sus sitios Webs preferidos. Se trata de un software que posee una interfaz gráfica compuesta básicamente de: botones de navegación, una barra de dirección, una barra de estado, que sirve para mostrar las páginas Web a las que se accede. [16]

2.2.8. Sistema de Gestión de Bases de Datos

Un sistema gestor de bases de datos o SGBD (aunque se suele utilizar más a menudo las siglas DBMS procedentes del inglés, Data Base Management System) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.[17]

2.2.9. Base de Datos

Es una colección de datos referentes a una organización estructurada según un modelo de datos de forma que refleja las relaciones y restricciones existentes entre los objetos del mundo real, y consigue independencia, integridad y seguridad de los datos.[18]

Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos se integran con una mínima cantidad de duplicidad. De este modo, la base de datos no pertenece a un solo departamento sino que se comparte por toda la organización. Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos.[19]

Se debe tener claro la diferencia entre Base de Datos y SGBD. La base de datos es el almacenamiento donde residen los datos. El SGBD es el encargado de manipular la información contenida en ese almacenamiento mediante operaciones de lectura/escritura sobre la misma. Además las bases de datos no sólo contendrán las tablas (ficheros) de datos, sino que también almacenará formularios (interfaces para edición de datos), consultas sobre los datos, e informes. [18]

2.3. Propuesta de Solución

El presente proyecto propone desarrollar un sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA, para facilitar la manipulación de la información, acceso a los datos, consultas oportunas, a la vez permitirá la confidencialidad, integridad, disponibilidad y seguridad de la información, ayudando en la gestión de programas de servicios comunitarios.

La generación de informes, permitirá obtener información completa de la gestión de programas de servicios comunitarios y ayudará a la posterior toma de decisiones, además al contar con un sistema automatizado el ahorro de tiempo y recursos serán evidentes.

CAPÍTULO 3

Metodología

El presente trabajo tendrá un enfoque de Investigación Aplicada ya que se orientará a los conocimientos adquiridos en la carrera universitaria y a la obtención de nuevos conocimientos para la aplicación en el desarrollo del sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA

3.1. Modalidad Básica de la Investigación

Investigación Aplicada

Se considera esta modalidad con la finalidad de aplicar los conocimientos obtenidos a lo largo de la carrera universitaria.

Investigación de Campo

Con el propósito de obtener los procesos que se realizan para la gestión de la información de los programas en el DIBESAU de la UTA, se realizará la investigación de campo donde se obtendrán datos reales.

Investigación Bibliográfica - Documentada

Se considera esta modalidad con el propósito de ampliar y profundizar el tema de investigación, con apoyo principalmente de libros, artículos, tesis desarrolladas en distintas universidades, por medios impresos, audiovisuales o electrónicos.

3.2. Recolección de Información

En el siguiente proyecto se realizó la recolección de información primaria mediante entrevistas con el personal del DIBESAU y observación de la gestión de programas comunitarios para lo cual se utilizó una guía de observación, así también la información secundaria se obtuvo a través de búsquedas en Internet de información

sobre el tema planteado, la utilización de documentación técnica, tesis, libros, con el fin de alcanzar los objetivos establecidos.

3.3. Población y Muestra

Para el proceso de investigación se trabajó con una población integrada por la directora y el técnico del DIBESAU, siendo la totalidad de la población dos personas. Debido al universo reducido no se ha sido necesario utilizar muestras de la población.

3.4. Procesamiento y Análisis de Datos

Para la recolección, procesamiento y análisis de la información se aplicará el siguiente procedimiento:

- Elaboración de los instrumentos de recolección de datos (entrevistas).
- Aplicación de las entrevistas. (Ver Anexo C)
- Análisis de los datos obtenidos de las entrevistas y la guía de observación.(Ver Anexo B)
- Interpretación de información en relación al problema y establecimiento de las respectivas conclusiones.

3.5. Desarrollo del Proyecto

- Recolección y análisis de la información.
- Definición ciclo de vida del software.
 - Modelos del ciclo de vida del software.
- Definición del modelo del ciclo de vida del software para el desarrollo del sistema.
- **Fase de Análisis**
 - Análisis y especificación de requerimientos.
 - Requerimientos funcionales.
 - ◇ Diagrama de casos de uso.
 - ◇ Especificación de casos de uso.

- Requerimientos no funcionales.
- Definición de la arquitectura de la aplicación Web.
- Conceptualización de herramientas.
- Selección de la tecnologías de acceso a datos
 - Cuadro Comparativo de las tecnologías de acceso a datos.
 - ◊ Análisis del cuadro comparativo.
- **Fase de Diseño**
 - Diseño de diagramas de secuencias.
 - Diseño del modelo de base de datos.
 - Realización del diccionario de datos.
 - Diseño de interfaces.
- **Fase de Implementación.**
 - Desarrollo de la aplicación Web.
- **Fase de Pruebas.**
 - Pruebas de funcionamiento.
 - Pruebas caja negra.
 - Pruebas caja blanca.
- **Fase de Mantenimiento.**
 - Implantación del sistema Web.

CAPÍTULO 4

Desarrollo de la Propuesta

Para concretar el desarrollo de la propuesta, se tomó como guía los objetivos específicos planteados, por cada objetivo específico se realizaron pasos concretos que se detallan a continuación:

4.1. Recolección y Análisis de la Información

Gracias a la apertura por parte de la directora la Dra. Eulalia Pino Loza y el Sr. Miguel López Técnico del DIBESAU de la UTA, se pudo obtener la información requerida, teniendo acceso a documentos físicos como el plan operativo anual; así también se pudo acompañar al Sr Miguel López Técnico Técnico DIBESAU para observar los distintos procesos que se realizan para llevar acabo la gestión de programas de servicios comunitarios, con lo cual se pudo realizar una guía de observación. (Ver Anexo B)

También se pudo obtener información por medio de las entrevistas que se aplicaron a los involucrados directos en la gestión de programas de servicios comunitarios como son, la Dra. Eulalia Pino Loza Directora y el Sr. Miguel López Técnico del DIBESAU de la UTA ya que, el manejo y la toma de decisiones de los programas de servicios comunitarios esta a cargo de la Directora y el seguimiento de los mismos esta a cargo del Técnico. El tipo de preguntas que se eligió para estas entrevistas fueron preguntas abiertas para poder obtener la información completa de los procesos. (Ver Anexo C)

Por medio de la guía de observación y de las entrevistas, se pudo evidenciar los procesos que se realizan en la gestión de programas de servicios comunitarios, así también los problemas que se generan al no contar con un sistema para gestión de programas, ya que la acumulación de la información genera trasapelado y más inconvenientes, es por ello, que al querer sacar conclusiones es complicado y sobre

todo la información no es completa, al no contar con un formato general para gestión de programas dificulta el análisis por parte de la directora para poder tomar decisiones.

4.2. Definición Ciclo de Vida del Software

El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (muere). También se denomina a veces paradigma. [20]

Entre las funciones que debe tener un ciclo de vida se pueden destacar:

- Determinar el orden de las fases del proceso de software.
- Establecer los criterios de transición para pasar de una fase a la siguiente.
- Definir las entradas y salidas de cada fase.
- Describir los estados por los que pasa el producto.
- Describir las actividades a realizar para transformar el producto.
- Definir un esquema que sirve como base para planificar, organizar, coordinar, desarrollar.[20]

4.2.1. Modelos del Ciclo de Vida del Software

La ingeniería del software establece y se vale de una serie de modelos que establecen y muestran las distintas etapas y estados por los que pasa un producto software, desde su concepción inicial, pasando por su desarrollo, puesta en marcha y posterior mantenimiento, hasta la retirada del producto. Los modelos de ciclo de vida del software describen las fases del ciclo de software y el orden en que se ejecutan las fases. Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociados entre estas etapas. [20]

Existen varias alternativas de modelos de ciclo de vida. A continuación se muestran algunos de los modelos tradicionales y más utilizados.

- Modelo en cascada: Es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior. El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo se ve

fluyendo hacia abajo (como una cascada) sobre las fases que componen el ciclo de vida. El modelo en cascada puede ser apropiado, en general, para proyectos estables (especialmente los proyectos con requisitos no cambiantes) y donde es posible y probable que los diseñadores predigan totalmente áreas de problema del sistema y produzcan un diseño correcto antes de que empiece la implementación. Funciona bien para proyectos pequeños donde los requisitos están bien entendidos.

- El modelo en v: es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Describe las actividades y resultados que han de ser producidos durante el desarrollo del producto. La parte izquierda de la v representa la descomposición de los requisitos y la creación de las especificaciones del sistema. El lado derecho de la v representa la integración de partes y su verificación. V significa “Validación y Verificación”.
- Modelo iterativo: Consiste en la interacción de varios ciclos de vida en cascada. Al final de cada interacción se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada interacción evalúa el producto y lo corrige o propone mejoras. Estas interacciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente.
- Modelo de desarrollo incremental: El modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.
- Modelo de Prototipos: El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente. El diseño rápido lleva a la construcción de un prototipo.
- Modelo en espiral: Este modelo de desarrollo combina las características del modelo de prototipos y el modelo en cascada. El modelo en espiral está pensado para proyectos largos, caros y complicados. Básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un modelo

en cascada, pero no necesariamente ha de ser así. [20]

4.3. Definición del Modelo del Ciclo de Vida del Software para el Desarrollo del Sistema

Al ver los distintos modelos del ciclo de vida del software, se ha optado por el modelo en cascada ya que se adapta a las características del desarrollo del sistema.

Así también a través de la recolección y análisis de la información por medio de las entrevistas y guía de observación, se pudo obtener información completa de la gestión de servicios comunitarios, permitiendo así tener claro los requerimientos para el sistema, se puede decir que los requerimientos son estables, con poca posibilidad de cambio, a la vez el sistema no es demasiado extenso, por lo cual el modelo en cascada se ajusta a dichas características.

Otro motivo por el cual se ha seleccionado este modelo es, debido a la facilidad de uso, a la organización que permite que el modelo sea fácil de gestionar ya que cada fase tiene un proceso de revisión y las fases son procesadas y completadas de una vez.

4.3.1. Modelo en Cascada

El modelo del ciclo de vida del software en cascada, es quizás el más ampliamente utilizado. En él, cada etapa deja el camino preparado para la siguiente, de forma que esta última no debe comenzar hasta que no ha acabado aquélla. De esta forma, se reduce mucho la complejidad de la gestión, ya que basta con no dar por terminada una etapa hasta que haya cumplido totalmente con sus objetivos. De esta forma, la siguiente puede apoyarse con total confianza en ella.[21] (Ver Figura 4.1)

4.3.1.1. Fases del Modelo en Cascada

- **Fase de análisis**

- Se analizan las necesidades de los usuarios finales del software a desarrollar para determinar qué objetivos debe cubrir. Es importante señalar que en esta etapa se deben verificar todo lo que se requiere en el sistema y será aquello lo que seguirá en las siguientes etapas, ya que no se pueden solicitar nuevos requisitos a mitad del proceso de elaboración del software.

- **Fase de diseño**

- Contiene la descripción de la estructura global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras. Se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

- **Fase de implementación**

- Es la fase de programación propiamente dicha. Aquí se desarrolla el código fuente.

- **Fase de pruebas**

- Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de ser puesto en explotación.

- **Fase de mantenimiento**

- El software obtenido se pone en producción. Es una de las fases finales del proyecto. En el desarrollo surgen cambios, para corregir errores o bien para introducir mejoras.[22]

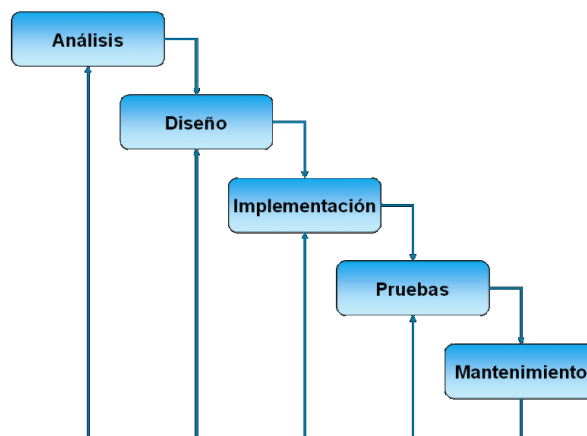


Figura 4.1: Modelo de ciclo de vida en cascada .

Fuente:

“EcuRed”,<http://www.ecured.cu/index.php/Archivo:Modelo-en-cascada.png>

4.4. Fase de Análisis

4.4.1. Análisis y Especificación de Requerimientos

Los requerimientos son el punto de partida para la construcción de un producto de software.

Los Requerimientos de software son un área de la Ingeniería de Software dedicada al estudio de la adquisición, análisis, especificación, validación y administración de los requerimientos o necesidades de un software. Es una fase fundamental para la construcción de software con calidad. Se entiende por calidad en un producto de software aquel que cumple con las necesidades del usuario.[23]

Por lo tanto una vez obtenida la información acerca de los procesos que se realizan para la gestión de programas de servicios comunitarios para el DIBESUA de la UTA, se pudo determinar los requerimientos necesarios para el desarrollo del sistema.

4.4.1.1. Requerimientos Funcionales

Los requerimientos funcionales incluyen las entradas y salidas de datos, los cálculos o las funciones del software.

La mejor manera de especificar los requerimientos funcionales son los casos de uso los que permiten conducir todo el proceso de desarrollo. Esta técnica se utiliza para identificar los requerimientos funcionales y a partir de ellos se diseña, implementa y prueba el software. Permite rastrear los requerimientos a través de todo el proceso de desarrollo hasta el producto terminado.[23]

Diagramas UML

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

El Lenguaje Unificado de Modelado (UML) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.[24]

- **Diagrama de Casos de Uso**

Los casos de uso proporcionan una manera incremental y modular de describir software, definen como utilizan el software sus usuarios.

El conjunto de casos de uso conforma el modelo de casos de uso que describe el comportamiento general del sistema. Los casos de uso proporcionan una

representación que puede ser fácilmente comprendida por todos los interesados. Se representan gráficamente con Diagramas de caso de uso de UML.[23]

Partiendo del concepto inicial de casos de uso y una vez identificados, los actores y los principales casos de usos para cada actor, se ha procedido a elaborar el diagrama de casos de uso que representa los procesos de gestión de programas de servicios comunitarios para el DIBESUA de la UTA.

Cada caso de uso cuenta con su respectiva especificación, es decir, quien puede realizar el proceso, que hace el proceso, sus precondiciones, sus pos condiciones, flujo principal y flujo secundario.

Se han determinado dos tipos de actores o usuarios: Administrador y Usuario.

- **Diagrama de Casos de Uso de los Procesos de Gestión de Programas de Servicios Comunitarios (Ver Figura 4.2)**
- **Especificación de Casos de Uso**

Caso de uso: Iniciar sesión	
Código:DB-0001	
Actores	<ul style="list-style-type: none"> ● Administrador. ● Usuario.
Descripción	Permite acceder al sistema para poder realizar tareas de acuerdo a sus privilegios.
Precondiciones	Estar registrado en el sistema.
Post condición	Ninguna.
Flujo Principal	<ol style="list-style-type: none"> 1. Ingresar en la URL. 2. Ingresar usuario y contraseña. 3. Presionar botón para ingresar.

Tabla 4.1: Caso de Uso: Iniciar Sesión

Elaborado por: Amalia Núñez

Caso de uso: Iniciar sesión	
Código:DB-0001	
Flujo Alternativo	<ol style="list-style-type: none"> 1. En caso de ingresar mal la información se mostrará el error: El Usuario o Contraseña es incorrecta. 2. En caso de dejar un campo vacío se mostrará el error: los campos de Usuario y Contraseña son obligatorios.

Tabla 4.2: Caso de Uso: Iniciar Sesión

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Cargos-Responsables	
Código:DB-0002	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El actor ingresa la información de cargos de los responsables.
Precondiciones	Estar identificado como administrador.
Post condición	Información del cargo ingresada correctamente.
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Datos Laborales. 2. Seleccionar Cargos. 3. Presionar el botón nuevo para ingresar el cargo. 4. Ingresar la información del cargo. 5. Guardar la información del cargo.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.3: Caso de Uso: Ingresar Cargos-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Modificar Cargos-Responsables	
Código:DB-0003	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El autor modifica los datos que han sido ingresados equivocadamente por alguna razón.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro del cargo debe existir.
Post condición	Información del cargo actualizada correctamente.
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Datos Laborales. 2. Seleccionar Cargos. 3. Buscar el registro a editar. 4. Seleccionar el registro a editar. 5. Presionar el botón de modificar. 6. Modificar la información requerida. 7. Guardar la información modificada.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.4: Caso de Uso: Modificar Cargos-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Cargos-Responsables	
Código:DB-0004	
Actores	<ul style="list-style-type: none"> • Administrador.

Tabla 4.5: Caso de Uso: Eliminar Cargos-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Cargos-Responsables	
Código:DB-0004	
Descripción	El autor elimina el registro seleccionado.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro del cargo debe existir. • El registro no debe estar utilizándose.
Post condición	Registro eliminado.
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Datos Laborales. 2. Seleccionar Cargos. 3. Buscar el registro a eliminar. 4. Seleccionar el registro a eliminar. 5. Presionar el botón de eliminar. 6. Eliminar registro.
Flujo Alternativo	Cancelar Eliminación.

Tabla 4.6: Caso de Uso: Eliminar Cargos-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Especialidades-Responsables	
Código:DB-0005	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El actor ingresa la información de especialidades de los responsables.
Precondiciones	Estar identificado como administrador.
Post condición	Información de especialidad ingresada correctamente.

Tabla 4.7: Caso de Uso: Ingresar Especialidades-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Especialidades-Responsables	
Código:DB-0005	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Responsables/Datos Laborales. 2. Seleccionar Especialidades. 3. Presionar el botón nuevo para ingresar la especialidad. 4. Ingresar la información de especialidad. 5. Guardar la información de especialidad.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.8: Caso de Uso: Ingresar Especialidades-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Modificar Especialidades-Responsables	
Código:DB-0006	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El autor modifica los datos que han sido ingresados equivocadamente por alguna razón.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de especialidad debe existir.
Post condición	Información de especialidad actualizada correctamente.

Tabla 4.9: Caso de Uso: Modificar Especialidades-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Modificar Especialidades-Responsables	
Código:DB-0006	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Responsables/Datos Laborales. 2. Seleccionar Especialidades. 3. Buscar el registro a editar. 4. Seleccionar el registro a editar. 5. Presionar el botón de modificar. 6. Modificar la información requerida. 7. Guardar la información modificada.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.10: Caso de Uso: Modificar Especialidades-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Especialidades-Responsables	
Código:DB-0007	
Actores	<ul style="list-style-type: none"> ● Administrador.
Descripción	El autor elimina el registro seleccionado.
Precondiciones	<ul style="list-style-type: none"> ● Estar identificado como administrador. ● El registro de especialidad debe existir. ● El registro no debe estar utilizándose.
Post condición	Registro eliminado.

Tabla 4.11: Caso de Uso: Eliminar Especialidades-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Especialidades-Responsables	
Código:DB-0007	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Responsables/Datos Laborales. 2. Seleccionar Especialidades. 3. Buscar el registro a eliminar. 4. Seleccionar el registro a eliminar. 5. Presionar el botón de eliminar. 6. Eliminar registro.
Flujo Alternativo	Cancelar Eliminación.

Tabla 4.12: Caso de Uso: Eliminar Especialidades-Responsables

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Responsables-Programas	
Código:DB-0008	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El actor ingresa la información del Responsable-Programa.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • Registro de Cargo ingresado. • Registro de especialidad ingresado
Post condición	Información del Responsable-Programa. ingresada correctamente.

Tabla 4.13: Caso de Uso: Ingresar Representantes-Programas

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Responsables-Programas	
Código:DB-0008	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Programas. 2. Seleccionar Responsables/Datos Personales. 3. Presionar el botón nuevo para ingresar la información. 4. Ingresar la información del representante-programa. 5. Guardar la información del responsable-programa.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.14: Caso de Uso: Ingresar Representantes-Programas

Elaborado por: Amalia Núñez

Caso de uso: Modificar Responsables-Programas	
Código:DB-0009	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor modifica los datos que han sido ingresados equivocadamente por alguna razón.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de Responsables/Datos Personales debe existir.
Post condición	Información del Responsable-Programa actualizada correctamente.

Tabla 4.15: Caso de Uso: Modificar Representantes-Programas

Elaborado por: Amalia Núñez

Caso de uso: Modificar Responsables-Programas	
Código:DB-0009	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Programas. 2. Seleccionar Responsables/Datos Personales. 3. Buscar el registro a editar. 4. Seleccionar el registro a editar. 5. Presionar el botón de modificar. 6. Modificar la información requerida. 7. Guardar la información modificada.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.16: Caso de Uso: Modificar Representantes-Programas

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Responsables-Programas	
Código:DB-0010	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor elimina el registro seleccionado.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de Responsables/Datos Personales debe existir. • El registro no debe estar utilizándose.
Post condición	Registro eliminado.

Tabla 4.17: Caso de Uso: Eliminar Representantes-Programas

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Responsables-Programas	
Código:DB-0010	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Programas. 2. Seleccionar Responsables/Datos Personales. 3. Buscar el registro a eliminar. 4. Seleccionar el registro a eliminar. 5. Presionar el botón de eliminar. 6. Eliminar registro.
Flujo Alternativo	Cancelar Eliminación.

Tabla 4.18: Caso de Uso: Eliminar Representantes-Programas

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Programas	
Código:DB-0011	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El actor ingresa la información del programa.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de Responsables/Datos Personales ingresado.
Post condición	Información del Programa ingresada correctamente.

Tabla 4.19: Caso de Uso: Ingresar Programas

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Programas	
Código:DB-0011	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Programas. 2. Seleccionar Nuevos Programas. 3. Presionar el botón nuevo para ingresar el programa. 4. Ingresar la información del programas. 5. Guardar la información del programa.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.20: Caso de Uso: Ingresar Programas

Elaborado por: Amalia Núñez

Caso de uso: Modificar Programas	
Código:DB-0012	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor modifica los datos que han sido ingresados equivocadamente por alguna razón.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro del programa debe existir.
Post condición	Información del Programa actualizada correctamente.

Tabla 4.21: Caso de Uso: Modificar Programas

Elaborado por: Amalia Núñez

Caso de uso: Modificar Programas	
Código:DB-0012	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Programas. 2. Seleccionar Nuevos Programas. 3. Buscar el registro a editar. 4. Seleccionar el registro a editar. 5. Presionar el botón de modificar. 6. Modificar la información requerida. 7. Guardar la información modificada.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.22: Caso de Uso: Modificar Programas

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Programas	
Código:DB-0013	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor elimina el registro seleccionado.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de Programa debe existir. • El registro no debe estar utilizándose.
Post condición	Registro eliminado.

Tabla 4.23: Caso de Uso: Eliminar Programas

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Programas	
Código:DB-0013	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Programas. 2. Seleccionar Nuevos Programas. 3. Buscar el registro a eliminar. 4. Seleccionar el registro a eliminar. 5. Presionar el botón de eliminar. 6. Eliminar registro.
Flujo Alternativo	Cancelar Eliminación.

Tabla 4.24: Caso de Uso: Eliminar Programas

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Responsables-Proyectos	
Código:DB-0014	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El actor ingresa la información del Responsable-Proyecto.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • Registro de Cargo ingresado. • Registro de especialidad ingresado
Post condición	Información del Responsable-Proyecto ingresada correctamente.

Tabla 4.25: Caso de Uso: Ingresar Representantes-Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Responsables-Proyectos	
Código:DB-0014	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Proyectos. 2. Seleccionar Responsables/Datos Personales. 3. Presionar el botón nuevo para ingresar la información. 4. Ingresar la información del representante-proyecto. 5. Guardar la información del responsable-proyecto.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.26: Caso de Uso: Ingresar Representantes-Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Modificar Responsables-Proyectos	
Código:DB-0015	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor modifica los datos que han sido ingresados equivocadamente por alguna razón.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de Responsables/Datos Personales debe existir.
Post condición	Información del Responsable-Proyecto actualizada correctamente.

Tabla 4.27: Caso de Uso: Modificar Representantes-Programas

Elaborado por: Amalia Núñez

Caso de uso: Modificar Responsables-Proyectos	
Código:DB-0015	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Proyectos. 2. Seleccionar Responsables/Datos Personales. 3. Buscar el registro a editar. 4. Seleccionar el registro a editar. 5. Presionar el botón de modificar. 6. Modificar la información requerida. 7. Guardar la información modificada.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.28: Caso de Uso: Modificar Representantes-Programas

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Responsables-Proyectos	
Código:DB-0016	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor elimina el registro seleccionado.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de Responsables/Datos Personales debe existir. • El registro no debe estar utilizándose.
Post condición	Registro eliminado.

Tabla 4.29: Caso de Uso: Eliminar Representantes-Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Responsables-Proyectos	
Código:DB-0016	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Proyectos. 2. Seleccionar Responsables/Datos Personales. 3. Buscar el registro a eliminar. 4. Seleccionar el registro a eliminar. 5. Presionar el botón de eliminar. 6. Eliminar registro.
Flujo Alternativo	Cancelar Eliminación.

Tabla 4.30: Caso de Uso: Eliminar Representantes-Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Proyectos	
Código:DB-0017	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El actor ingresa la información del proyecto.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro del programa debe existir.
Post condición	Información del Proyecto ingresada correctamente.

Tabla 4.31: Caso de Uso: Ingresar Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Proyectos	
Código:DB-0017	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Proyectos. 2. Seleccionar Nuevos Proyectos. 3. Presionar el botón nuevo para ingresar el proyecto. 4. Ingresar la información del proyecto. 5. Guardar la información del proyecto.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.32: Caso de Uso: Ingresar Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Modificar Proyectos	
Código:DB-0018	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor modifica los datos que han sido ingresados equivocadamente por alguna razón.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro del Proyecto debe existir.
Post condición	Información del Proyecto actualizada correctamente.

Tabla 4.33: Caso de Uso: Modificar Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Modificar Proyectos	
Código:DB-0018	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Proyectos. 2. Seleccionar Nuevos Proyectos. 3. Buscar el registro a editar. 4. Seleccionar el registro a editar. 5. Presionar el botón de modificar. 6. Modificar la información requerida. 7. Guardar la información modificada.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.34: Caso de Uso: Modificar Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Proyectos	
Código:DB-0019	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor elimina el registro seleccionado.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • Registro del Proyecto debe existir. • El registro no debe estar utilizándose.
Post condición	Registro eliminado.

Tabla 4.35: Caso de Uso: Eliminar Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Proyectos	
Código:DB-0019	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el Menú Proyectos. 2. Seleccionar Nuevos Proyectos. 3. Buscar el registro a eliminar. 4. Seleccionar el registro a eliminar. 5. Presionar el botón de eliminar. 6. Eliminar registro.
Flujo Alternativo	Cancelar Eliminación.

Tabla 4.36: Caso de Uso: Eliminar Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Asignación de responsables a Proyectos	
Código:DB-0020	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El actor ingresa la información de la asignación de los responsables al proyecto.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro del proyecto debe existir.
Post condición	Información de la asignación de los responsables al proyecto ingresada correctamente.

Tabla 4.37: Caso de Uso: Ingresar Asignación de responsables a Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Asignación de responsables a Proyectos	
Código:DB-0020	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Proyectos. 2. Seleccionar Asignar responsables a proyecto. 3. Presionar el botón nuevo para ingresar el proyecto. 4. Ingresar la información requerida. 5. Guardar la información de la asignación de los responsables al proyecto.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.38: Caso de Uso: Ingresar Asignación de responsables a Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Modificar Asignación de responsables a Proyectos	
Código:DB-0021	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor modifica los datos que han sido ingresados equivocadamente por alguna razón.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de Asignación de responsables a Proyecto debe existir.
Post condición	Información de Asignación de responsables a Proyecto actualizada correctamente.

Tabla 4.39: Caso de Uso: Modificar Asignación de responsables a Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Modificar Asignación de responsables a Proyectos	
Código:DB-0021	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Proyectos. 2. Seleccionar Asignar responsables a proyecto. 3. Buscar el registro a editar. 4. Seleccionar el registro a editar. 5. Presionar el botón de modificar. 6. Modificar la información requerida. 7. Guardar la información modificada.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.40: Caso de Uso: Modificar Asignación de responsables a Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Asignación de responsables a Proyectos	
Código:DB-0022	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor elimina el registro seleccionado.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • Registro de Asignación de responsables a Proyectos debe existir. • El registro no debe estar utilizándose.
Post condición	Registro eliminado.

Tabla 4.41: Caso de Uso: Eliminar Asignación de responsables a Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Asignación de responsables a Proyectos	
Código:DB-0022	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Proyectos. 2. Seleccionar Asignar responsables a Proyecto. 3. Buscar el registro a eliminar. 4. Seleccionar el registro a eliminar. 5. Presionar el botón de eliminar. 6. Eliminar registro.
Flujo Alternativo	Cancelar Eliminación.

Tabla 4.42: Caso de Uso: Eliminar Asignación de responsables a Proyectos

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Actividades	
Código:DB-0023	
Actores	<ul style="list-style-type: none"> • Administrador
Descripción	El actor ingresa la información de la actividad.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro del proyecto debe existir.
Post condición	Información de la actividad ingresada correctamente.

Tabla 4.43: Caso de Uso: Ingresar Actividades

Elaborado por: Amalia Núñez

Caso de uso: Ingresar Actividades	
Código:DB-0023	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Actividades 2. Seleccionar Nuevas Actividades. 3. Presionar el botón nuevo para ingresar la actividad. 4. Ingresar la información de la actividad. 5. Guardar la información de la actividad.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.44: Caso de Uso: Ingresar Actividades

Elaborado por: Amalia Núñez

Caso de uso: Modificar Actividades	
Código:DB-0024	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor modifica los datos que han sido ingresados equivocadamente por alguna razón.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • El registro de la actividad debe existir.
Post condición	Información de la actividad actualizada correctamente.

Tabla 4.45: Caso de Uso: Modificar Actividades

Elaborado por: Amalia Núñez

Caso de uso: Modificar Actividades	
Código:DB-0024	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Actividades. 2. Seleccionar Nuevas Actividades. 3. Buscar el registro a editar. 4. Seleccionar el registro a editar. 5. Presionar el botón de modificar. 6. Modificar la información requerida. 7. Guardar la información modificada.
Flujo Alternativo	En caso de ingresar mal la información se mostrará el error: Los campos con (*) son obligatorios.

Tabla 4.46: Caso de Uso: Modificar Actividades

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Actividades	
Código:DB-0025	
Actores	<ul style="list-style-type: none"> • Administrador.
Descripción	El autor elimina el registro seleccionado.
Precondiciones	<ul style="list-style-type: none"> • Estar identificado como administrador. • Registro de la Actividad debe existir. • El registro no debe estar utilizándose.
Post condición	Registro eliminado.

Tabla 4.47: Caso de Uso: Eliminar Actividades

Elaborado por: Amalia Núñez

Caso de uso: Eliminar Actividades	
Código:DB-0025	
Flujo Principal	<ol style="list-style-type: none"> 1. Seleccionar en el menú Actividades. 2. Seleccionar Nuevas Actividades. 3. Buscar el registro a eliminar. 4. Seleccionar el registro a eliminar. 5. Presionar el botón de eliminar. 6. Eliminar registro.
Flujo Alternativo	Cancelar Eliminación.

Tabla 4.48: Caso de Uso: Eliminar Actividades

Elaborado por: Amalia Núñez

4.4.1.2. Requerimientos no Funcionales

Los requerimientos no funcionales tienen que ver con los atributos o características que el cliente solicita para el funcionamiento del software.

Los requerimientos no funcionales pueden ser:

- Interfaz con el usuario: descripción de las características que permitan al software apoyar al usuario en sus tareas.
- Confiabilidad: solicitud del desempeño respecto a seguridad, tolerancia a fallas y su recuperación.
- Eficiencia: límites de tiempo de respuesta.
- Restricciones de diseño y construcción: las que imponga el cliente. [23]

Una vez identificadas las características de funcionamiento que el cliente requiere en el sistema para gestión de programas de servicios comunitarios para el DIBESUA de la UTA , se establecen los requerimientos no funcionales para dicho sistema.

- **Interfaz con el usuario**
 - La interfaz que se muestre al usuario debe ser amigable e intuitiva.
- **Confiabilidad**
 - El sistema deberá de tener una alta confiabilidad e integridad con los datos.

- Para poder ingresar, modificar, eliminar datos, debe contar con los permisos de administrador.
 - El sistema debe permitir el ingreso a los usuarios registrados con sus respectivos privilegios, es decir el usuario debe tener un nombre de usuario y contraseña previamente ingresadas por el administrador.
- **Eficiencia**
 - El sistema debe proporcionar un tiempo de respuesta rápido.
- **Restricciones de diseño y construcción**
 - El sistema de software será construido para funcionar en ambiente Web.
 - El sistema debe ser desarrollado en Visual Studio 2010 en vista que la Universidad Técnica de Ambato posee las licencias y trabaja con el mismo.
 - La base de datos del sistema deberá estar construida con el gestor de base de bases de datos SQL Server ya que la Universidad Técnica de Ambato cuenta con las licencias y trabaja con dicho gestor.
 - Para el desarrollo de este sistema se utilizarán las siguientes herramientas:
 - Star UML para modelado del sistema.
 - ASP.NET para el desarrollo del sitio Web.
 - Visual Studio 2010 (C#) para la codificación.
 - SQL-Server para la creación de la Base de Datos.
 - Internet Information Server para la publicación del sitio Web.
 - LyX para hacer la documentación.



Figura 4.2: Diagrama de Caso de Uso de los Procesos de Gestión de Servicios Comunitarios

Elaborado por: Amalia Núñez

4.4.2. Definición de la Arquitectura de la Aplicación Web

La arquitectura que se va a utilizar para la implantación de la aplicación Web es la arquitectura clásica cliente/servidor, ya que una aplicación Web se basa en la arquitectura cliente/servidor y cumple con los componentes necesarios que se desean,

como es por un lado el cliente (el navegador) y por otro lado el servidor (el servidor Web).

Existen diversas variantes de la arquitectura básica según como se implementen las diferentes funcionalidades de la parte servidor.

En vista que la implantación será localmente en el DIBESAU de la UTA y esta dependencia solo cuenta con un servidor, se ha decidido utilizar todo en un servidor, es decir un único ordenador alojará el servicio de HTTP (Hypertext Transfer Protocol), la lógica de negocio y la lógica de datos y los datos.

El software que ofrece el servicio de HTTP gestiona también la lógica de negocio. Esta arquitectura es usada en las tecnologías como ASP (Active Server Pages) Y PHP (Hypertext Preprocessor).

La arquitectura de programación elegida es en capas, por las facilidades de programación que brinda.

4.4.2.1. Arquitectura Cliente/Servidor

La arquitectura cliente/servidor es una arquitectura de comunicación de computadores en el cual el computador cliente solicita servicios al computador servidor por medio de mensajes.

La diferencia entre el cliente y el servidor es que el cliente es el que inicia el contacto y el servidor es el que responde a dicha solicitud de conexión. [25]

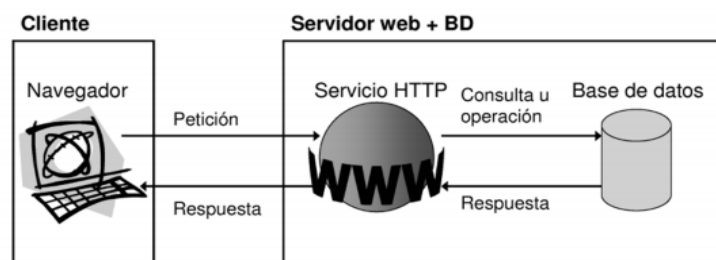


Figura 4.3: Arquitectura Cliente/Servidor.

Fuente: “TRIA – Tecnólogo Informático”,

<http://www.fing.edu.uy/tecnoinf/mvd/cursos/ria/material/teorico/ria-01-introduccion.pdf>

Características

En la arquitectura cliente/servidor el remitente de una solicitud es conocido como cliente. Sus características son:

- Es el que inicia solicitudes o peticiones. Tiene, por tanto, un papel activo en la comunicación.
 - Espera y recibe las respuestas del servidor.
 - Por lo general, puede conectarse a varios servidores a la vez.
 - Normalmente, interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.
 - Al contratar un servicio de red, se debe tener en cuenta la velocidad de conexión que se le otorga al cliente y el tipo de cable que utiliza.
- Al receptor de la solicitud enviada por el cliente se conoce como servidor. Sus características son:
 - Al iniciarse espera a que le lleguen las solicitudes de los clientes. Desempeñan entonces un papel pasivo en la comunicación.
 - Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
 - Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
 - No es frecuente que interactúen directamente con los usuarios finales. [26]

■ Ventajas

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor, de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar o incluso trasladar un servidor, mientras que sus clientes

no se verán afectados por ese cambio. Esta independencia de los cambios también se conoce como encapsulación.

- Tecnologías: existen algunas suficientemente desarrolladas, diseñadas para el paradigma de cliente/servidor, que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz y la facilidad de empleo.[25]

■ Desventajas

- La congestión del tráfico ha sido siempre un problema en el paradigma de cliente-servidor. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). [25]

Back - End y Front -End

En el diseño de software, Front-End es la parte de un sistema de software que interactúa directamente con el usuario; mientras que Back-End comprende los componentes que procesan la salida del Front-End. La separación de los sistemas de software en "Front-End" y "Back-End", es una abstracción que sirve para mantener separadas las diferentes partes de un sistema.

- El Front-End en diseño de software y desarrollo Web hace referencia a todas aquellas tecnologías que corren del lado del cliente, visualización del usuario navegante o, dicho de otra manera, es la parte que interactúa con los usuarios. De una manera práctica, el Front-End sería todos esos formularios de las Webs, las aplicaciones que se manejan y todas esas cosas que tengan que realizar los usuarios.
- El Back-End es la parte que procesa la entrada de datos que se efectuó desde el Front-End es decir es lo que se encuentra del lado del servidor, son los procesos que utiliza el administrador del sitio con sus respectivos sistemas para resolver las peticiones de los usuarios. De esta manera en manera conjunta el Front-End y el Back-End interactúan en un sistema Web o software para resolver las necesidades de los usuarios.[27]

A continuación se detallan las herramientas utilizadas tanto para el Back - End como para el Front - End.

4.4.3. Conceptualización de Herramientas

Back-End

4.4.3.1. Sistema de Gestión de Base de Datos (SGBD)

Hay varios sistemas de gestión de base de datos, se ha tomado como referencia los más populares como son MySQL, PostgreSQL y SQL-Server para realizar un cuadro comparativo con sus principales características:

Característica	MySQL	PostgreSQL	SQL-Server
Plataformas	Windows, Linux, Unix, Mac	Windows, Linux, Unix, Mac	Windows
Open Source	Si	Si	No
Instalación	Medio	Fácil	Medio
Métodos de Conexión	JDBC, ODBC, ADO.NET	JDBC, ODBC, ADO.NET	ODBC, OLE DB, ADO.NET, JDBC
Seguridades	Media	Alta	Alta
Soporte de Transacciones	Si	Si	Si
Vistas Indexadas	No	Si	Si
Vistas actualizables	Si	Si	Si
Vista gráfica del diseño	No	No	Si
Servicio de reportes	No	Si	Si
Posibilidad de elegir diferentes formas de almacenamiento	Si	Si	No
Orientación a objetos	No	Si	Si
Robusto	No	Si	Si
Encriptación de datos internamente	No	Si	Si
Procedimientos almacenados	Si	Si	Si
Triggers	Si	Si	Si
Replicación	Si	Si	Si
Documentación	Si	Si	Si

Tabla 4.49: Cuadro Comparativo Sistemas de Gestión de Base de Datos.

Análisis del Cuadro Comparativo

Al Analizar el cuadro comparativo se puede concluir que los sistemas de gestión de base de datos PostgreSQL y SQL-server son los mejores ya que cumplen con la

mayoría de las características establecidas, con un alto rendimiento en, confiabilidad, integridad, seguridad, robustez, soporte de transacciones y disponibilidad de la información.

En vista de los resultados obtenidos de la comparación entre MySQL, PostgreSQL y SQL-Server, se decide la utilización del SGBD SQL-server, el principal motivo que indujo a tomar dicha decisión es que la Universidad Técnica de Ambato cuenta con las licencias de este SDBG, otro motivo es la gran compatibilidad entre SQL-Server y ASP.NET, además que es un SGBD muy eficiente, seguro y con un gran desempeño en el manejo de transacciones de la información.

Microsoft SQL Server

Es un sistema para la gestión de bases de datos creado por Microsoft, el mismo se basa en el modelo relacional.

■ Características

- Soporte de transacciones.
 - Escalabilidad, estabilidad y seguridad a procedimientos almacenados.
 - Incluye un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
 - Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
 - Permite administrar información de otros servidores de datos. [28]

Front - End

4.4.3.2. Herramientas de Programación

Hoy en día existen múltiples herramientas de programación se ha decidido tomar como referencia las dos más populares y las que se utilizan para el desarrollo Web en la UTA como son, PHP y ASP.NET para realizar un cuadro comparativo que permita establecer cual de ellas es la más adecuada para el desarrollo del sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.

Característica	PHP	ASP.NET
Plataformas	Windows, Linux, Unix, Mac	Windows
Open Source	Si	No
Instalación	Medio	Medio
Framework	No	Si
Programación orientada a objetos	Deficiente	Buena
Manejo de Errores	Medio	Alto
IDE Visual	La mayoría de los IDEs PHP, requieren gran cantidad de complementos con el fin de añadir funciones similares a Visual Studio.	Las aplicaciones ASP.Net pueden ser desarrolladas con el impresionante IDE Visual Studio.Net que ofrece amplia gama de características, que hacen mucho más fácil la codificación y el desarrollo productivo.
Soporte nativo para AJAX	No	ASP.Net incorpora soporte para AJAX desde Net Framework 3.5
Servidor	Apache	IIS
Memoria Requerida	128 o más	128 o más
Escalabilidad	Alta	Alta
Facilidad de aprendizaje	Medio	Alto
Manejo de excepciones	Si	Si
División entre la capa de aplicación o diseño y el código	No	Si
Controles de usuario	No	Si
Ejecución de código	Php es un lenguaje interpretado, lo que significa que se guarda como el código escrito y se ejecuta directamente desde el mismo.	Asp.Net es un lenguaje optimizado y compilado, es decir, el código que escriba se reduce a un conjunto de instrucciones específicas de la máquina antes de ser guardado como un archivo ejecutable
Documentación	Si	Si

Tabla 4.50: Cuadro Comparativo Herramientas de Programación.

Análisis del Cuadro Comparativo

Observando el cuadro anterior se puede evidenciar que tanto PHP como ASP.NET son muy eficientes, PHP cuenta con gran popularidad por ser un software libre,

ASP.NET cuenta con la facilidad de aprendizaje ya que cuenta con el IDE Visual Studio que brinda un ambiente de desarrollo amigable.

Por otro lado ASP.NET brinda grandes características como el soporte nativo de AJAX que facilita mucho el trabajo del desarrollo Web a través de sus componentes, así también los controles de usuarios que se pueden personalizar.

En vista de las grandes capacidades de ASP.NET y sobre todo que la Universidad Técnica de Ambato cuenta con las licencias correspondientes, se ha optado por la utilización de ASP.NET para el desarrollo del sistema para gestión de programas de servicios comunitarios para el DIBESAU de la UTA.

- **ASP.NET**

Es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc.[29]

- **Visual Studio**

Es un conjunto de herramientas de desarrollo basadas en componentes y otras tecnologías para compilar aplicaciones eficaces de alto rendimiento. Además, Visual Studio está optimizado para el diseño, el desarrollo y la implementación en equipo de soluciones empresariales.[30]

- **Lenguaje de Programación C Sharp (C#)**

C# o C Sharp es un lenguaje de programación que está incluido en la Plataforma .NET y corre en el Lenguaje Común en Tiempo de Ejecución (CLR, Common Language Runtime). El primer lenguaje en importancia para el CLR es C#, mucho de lo que soporta la Plataforma .NET está escrito en C#.

C# deriva de C y C++, es moderno, simple y enteramente orientado a objetos, simplifica y moderniza a C++ en las áreas de clases, namespaces, sobrecarga de métodos y manejo de excepciones. Se eliminó la complejidad de C++ para hacerlo más fácil de utilizar y menos propenso a errores. C# es “case sensitive”, es decir, que distingue mayúsculas de minúsculas. HolaMundo es diferente a holamundo. [31]

- **.Net Framework**

Es un componente integral de Windows que admite la compilación y la ejecución de la siguiente generación de aplicaciones y servicios Web. Los componentes clave de .NET Framework son Common Language Runtime (CLR) y la biblioteca de clases .NET Framework, que incluye ADO.NET, ASP.NET, formularios Windows Forms y Windows Presentation Foundation (WPF).

.NET Framework proporciona un entorno de ejecución administrado, un desarrollo e implementación simplificada y la integración con una gran variedad de lenguajes de programación. [32]

- **Common Language Runtime (CLR)**

.NET Framework proporciona un entorno en tiempo de ejecución denominado Common Language Runtime, que ejecuta el código y proporciona servicios que facilitan el proceso de desarrollo.

Los compiladores y las herramientas exponen la funcionalidad de Common Language Runtime y permiten escribir código con las ventajas que proporciona este entorno de ejecución administrado. El código desarrollado con un compilador de lenguaje orientado al tiempo de ejecución se denomina código administrado.

Este código se beneficia de características como: la integración entre lenguajes, el control de excepciones entre lenguajes, la seguridad mejorada, la compatibilidad con la implementación y las versiones, un modelo simplificado de interacción y servicios de generación de perfiles y depuración. [33]

- **AJAX**

AJAX son las siglas de Asynchronous JavaScript And XML. No es un lenguaje de programación sino un conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que permite hacer páginas más interactivas. La característica fundamental de AJAX es permitir actualizar parte de una página con información que se encuentra en el servidor sin tener que refrescar completamente la página. De modo similar se puede enviar información al servidor. [34]

- **ASP.NET AJAX Control Toolkit**

Nace como un proyecto conjunto entre la comunidad de programadores y Microsoft. Está desarrollado en base a ASP.NET AJAX y contiene una

serie de controles Web y extendedores con los que se puede utilizar las avanzadas características de ASP.NET AJAX sin más que un arrastre de ratón.

Del mismo modo, con su descarga gratuita se dispone de ejemplos de uso, así como del propio código fuente de los controles. [35]

■ **Componente de Visualización**

El componente ReportViewer se va a encargar de contener el informe, el cual se carga tanto dinámica, como estáticamente.

El control Web del visor de informes requiere un System.Web.UI.ScriptManager en el formulario Web, ya que internamente, el control utiliza llamadas Ajax para generar el informe.

● **Propiedades de un ReportViewer**

Entre las propiedades del ReportViewer se puede destacar las siguientes:

- SizeToReportContent: Determina si el área del informe tiene un tamaño fijo o si equivale al tamaño del contenido del informe.
- Propiedades de la categoría “Barra de herramientas”: Hace que se muestre u oculte cualquier botón del ReportViewer (flechas de navegación, botón imprimir, ...).
- AsyncRendering: Determina si el informe se representa asincrónicamente a partir del resto de la página. [36]

4.4.4. **Selección de la Tecnología de Acceso a Datos**

La mayoría de las aplicaciones necesitan algún mecanismo de acceso a datos. De hecho, en la mayoría de los casos de desarrollo de sistemas es a lo que más tiempo se le dedica.

En ASP.NET existen diversas tecnologías para conseguir el acceso a datos, a lo largo del tiempo estas han ido evolucionando.

.NET Framework 1.1 incluyó la tecnología de acceso a datos denominada ADO.NET, una mejora de ADO, en la versión 3.5 de .NET Framework se incluyó la herramienta de O/RM (Object Relational Mapping) conocida como "LINQ to SQL", más adelante, con la versión 4.0 de .NET Framework se incluyó ADO.NET Entity Framework y su LINQ to Entities.[37]

Al ver las distintas tecnologías de acceso a datos que posee ASP.NET, se realizará una comparativa entre las tres tecnologías de acceso a datos más importantes que posee ASP.NET, como son, el clásico ADO.NET, LINQ to SQL y ADO.NET Entity Framework para determinar cual de ellas es la mas idónea para realizar el acceso a datos del sistema.

4.4.4.1. ADO.NET

ADO.NET es la siguiente generación ActiveX (ADO). Es un modelo para acceder a datos generados con base en la escalabilidad, XML y la falta de estado de Web, ADO.NET proporciona una interfaz para todos los orígenes de datos de OLE DB y le permite conectarse a ellos, recuperarlos y actualizarlos.

Puede usarlos ya sea que esté en un entorno remoto, utilizando una aplicación distribuida, o datos desconectados. ADO.NET Forma parte integral de .NET Framework.[38]

Componentes de ADO.NET

Los componentes de ADO.NET están diseñados para separar el acceso a datos de la manipulación de datos. ADO.NET tiene dos componentes principales que cumplen esta función: el DataSet y el proveedor de datos de .NET Framework.

- El DataSet de ADO.NET es el componente central de la arquitectura sin conexión de ADO.NET. El DataSet está expresamente diseñado para el acceso a datos independientemente del origen de datos. Como resultado, se puede utilizar con múltiples y distintos orígenes de datos, con datos XML o para administrar datos locales de la aplicación.
- Proveedor de datos de .NET Framework, cuyos componentes están diseñados expresamente para la manipulación de datos y para el acceso rápido a datos de avance de sólo lectura. El objeto Connection proporciona conectividad con un origen de datos.

ADO.NET Arquitectura de ADO.NET

En la figura 4.4 se observa que el Data Provider (Proveedor de datos) de .NET está conectado con la base de datos. A su vez, el DataSet se comunica con el Data Provider para intercambiar los datos que se transmiten y reciben en XML, hacia y desde, por ejemplo un Servicio Web.

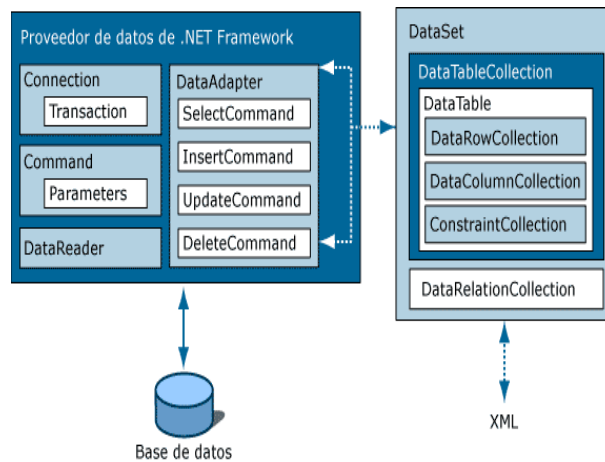


Figura 4.4: Arquitectura de ADO.NET.

Fuente: “Microsoft”,

[https://msdn.microsoft.com/es-es/library/aa719511\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa719511(v=vs.71).aspx)

Proveedores de Datos para ADO.NET

Microsoft ofrece dos proveedores administrados con ADO.NET:

El proveedor administrado SQL y el proveedor administrado OLE DB, el primero se usa unicamente con Microsoft SQL Server y ofrece todos los métodos para la comunicación entre SQL Server y el DataSet.

El segundo OLE DB, interviene en la comunicación entre un DataSet y cualquier origen de datos compatible con OLE DB.[38]

■ Ventajas

● Interoperabilidad

- Las aplicaciones ADO.NET pueden aprovechar la flexibilidad y la amplia aceptación de XML. Dado que XML es el formato de transmisión de conjuntos de datos a través de la red, cualquier componente que pueda leer el formato XML podrá procesar los datos.

● Escalabilidad

- Para facilitar la escalabilidad, ADO.NET anima a los programadores a ahorrar recursos limitados. Las aplicaciones ADO.NET utilizan un acceso desconectado a los datos, por lo que no retienen bloques ni conexiones activas con bases de datos durante largos períodos de tiempo.

- **Mantenibilidad**

- A lo largo de la vida de un sistema implementado es posible hacer cambios modestos, pero raramente se intenta hacer cambios importantes.

- **Programabilidad**

- En Visual Studio, los componentes de datos ADO.NET encapsulan funcionalidad de acceso a datos de diversas formas que ayudan a programar de modo más rápido y con menos errores.

- **Rendimiento**

- Para las aplicaciones desconectadas, los conjuntos de datos ADO.NET ofrecen ventajas de rendimiento.[39]

- **Desventajas**

- La capa de acceso a datos DAL (DAtaAccess Layer) se la realiza manualmente, por lo que incrementa la cantidad de código.
- La configuración de la capa de acceso a datos requiere en la mayoría de los casos mucho trabajo.
- DataSet tiene funciones de consulta limitadas.
- En un escenario conectado una conexión constante consume muchos recursos.
- En un escenario conectado el acceso a datos se ve limitado por las múltiples conexiones que tiene el servidor.
- En un escenario desconectado al sincronizarse, los datos pueden haber conflictos los cuales se deberán resolver de manera manual.

4.4.4.2. LINQ to SQL

En la versión 3.5 de .NET Framework se incluyó la herramienta de O/RM (Object Relational Mapping) conocida como "LINQ to SQL".

El mapeo objeto-relacional es una técnica de programación para convertir datos del sistema de tipos, utilizado en un lenguaje de programación orientado a objetos, al utilizado en una base de datos relacional. En la práctica esto crea una base de datos virtual orientada a objetos sobre la base de datos relacional. [40]

Las bases de datos relacionales solo permiten guardar tipos de datos primitivos (enteros, cadenas de texto, etc.) por lo que no se pueden guardar de forma directa los objetos de la aplicación en las tablas, sino que estos se deben de convertir antes en registros, que por lo general afectan a varias tablas.

En el momento de volver a recuperar los datos, hay que hacer el proceso contrario, se deben convertir los registros en objetos.

Es entonces cuando ORM cobra importancia, ya que se encarga de forma automática de convertir los objetos en registros y viceversa, simulando así tener una base de datos orientada a objetos.[41]

LINQ to SQL puede modelar una base de datos relacional con clases .NET de forma automática lo que permite seleccionar, editar, insertar o eliminar datos tan solo editando las clases y colecciones generadas.

Para poder trabajar con estas clases se añadió soporte para lenguaje "LINQ" (Language Integrated Query) que no es más que un modo de expresar consultas similares al SQL normal contra cualquier colección (ya sea generado por el modelador de LINQ to SQL o bien creada manualmente).

LINQ incluye tipado, comprobación de las consultas en tiempo de compilación, expresiones lambda y tipos anónimos e intellisense lo que lo convierte en un lenguaje para consultas mucho más potente que el SQL clásico.[37]

Componentes de LINQ

Dependiendo de la fuente de datos a trabajar, es el componente LINQ a utilizar, los componentes se agrupan en:

■ LINQ to SQL

- Es el conjunto de clases, estructuras, interfaces y enumeraciones utilizadas para escribir consultas a bases de datos relacionales como PostgreSQL, SQL Server o MySQL. [42]
 - LINQ to SQL proporciona una infraestructura en tiempo de ejecución para administrar los datos relacionales como objetos.
 - En LINQ to SQL, el modelo de datos de una base de datos relacional se asigna a un modelo de objetos expresado en el lenguaje de programación del desarrollador.
 - Al ejecutar la aplicación, LINQ to SQL convierte las consultas integradas en el lenguaje del modelo de objetos a SQL y las envía a la base de datos para su ejecución. Cuando la base de datos devuelve

los resultados, LINQ to SQL los vuelve a convertir en objetos que se pueden manipular.[43]

- **LINQ to Objects**

- Es la API predeterminada de LINQ y permite escribir consultas para arreglos, estructuras y colecciones de objetos en memoria.

- **LINQ to XML**

- Proporciona la habilidad de escribir consultas para procesar fuentes de datos XML.[42]

- **LINQ to DataSet**

- DataSet es uno de los componentes más utilizados en ADO.NET y es un elemento clave del modelo de programación desconectado en el que se basa ADO.NET. A pesar de este punto destacado, DataSet tiene funciones de consulta limitadas.
- LINQ to DataSet permite integrar funciones de consulta más completas en DataSet mediante la misma funcionalidad de consulta que está disponible para muchos otros orígenes de datos.[43]

Sintaxis y Operadores de las consultas en LINQ

Básicamente la sintaxis de las consultas en LINQ es similar a las consultas SQL, utilizando las palabras reservadas select, from, where, orderby join. [42]

Aquí a diferencia de las consultas SQL la palabra reservada from va al comienzo de la consulta y al final se emplea la palabra reservada select, como en el código siguiente:

```
1 var fromQuery = from a in lAlmacenes select a;
```

Similar a SQL tenemos la palabra reservada where , como en el código siguiente:

```
1 var whereQuery = from a in lAlmacenes where a.Hospital.ID == 2
   select a;
```

El uso de orderby y descending se ejemplifica en los códigos siguientes:

```
1 var orderbyQuery = from a in lAlmacenes orderby a.NombreA select a;
2 var orderbyQueryDesc = from a in lAlmacenes orderby a.NombreA
   descending select a;
```


Continuando con la similitud con SQL no debe faltar la palabra reservada join, como en el siguiente código:

```
1 var joinQuery = from r in lArticulos join a in lAlmacenes on r.Almacen.  
    IDAlmacen equals a.IDAlmacen select new {r.Nombre,a.NombreA };
```

■ Ventajas

- El Diseñador relacional de objetos (Diseñador R/O) proporciona una superficie de diseño visual para crear y editar las clases de LINQ to SQL (clases de entidad) basadas en los objetos de una base de datos. Mediante LINQ to SQL, se puede obtener acceso a las bases de datos SQL con tecnología LINQ.
- LINQ proporcionar un modelo coherente para trabajar con los datos de varios tipos de formatos y orígenes de datos. [44]
- En una consulta LINQ, siempre se trabaja con objetos. [44]
- Se utilizan los mismos modelos de codificación básicos para consultar y transformar datos de documentos XML, bases de datos SQL, conjuntos de datos ADO.NET, colecciones .NET y cualquier otro formato para el que haya disponible un proveedor LINQ.[44]
- LINQ to DataSet facilita y acelera las consultas en datos almacenados en caché en un objeto DataSet. [45]
- LINQ to DataSet simplifica la consulta permitiendo a los desarrolladores escribir consultas a partir del lenguaje de programación mismo, en lugar de utilizar un lenguaje de consulta diferente. [45]
- La creación de DAL (DATAAccess Layer) se vuelve una tarea sencilla debido a que no se utiliza, el lenguaje SQL, el cual es reemplazado por peticiones LINQ.[46]
- Linq to SQL gestiona la base de datos.[46]
- El mapeo O/RM permite generar objetos a partir de un modelo de base de datos.[46]

■ Desventajas

- LINQ to SQL sólo permite conectarse a SQL Server.
- LINQ to SQL mapea cada tabla/vista como una clase o “entidad” sin permitir herencia.
- LINQ to SQL sólo gestiona la base de datos y no gestionan las relaciones n-n (muchos-a-muchos).

4.4.4.3. Entity Framework

Es un conjunto de tecnologías en ADO.NET que dan soporte al desarrollo de software orientado a datos. Es un ORM (Mapeo Objeto-Relacional), que permite a los desarrolladores .NET tener mecanismos para acceder y guardar datos en una base de datos. [47]

ADO.NET Entity Framework permite a los desarrolladores crear aplicaciones de acceso a datos programando con un modelo de aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y el mantenimiento necesarios para las aplicaciones orientadas a datos.[48]

Entity Entity Framework (EF) abstrae la estructura de la base de datos y todo el acceso y almacenamiento de datos se realiza frente a un modelo conceptual de datos que refleja los objetos de negocio. Los items usados para representar el modelo conceptual se conocen como Entidades.[49]

Con ADO.NET Entity Framework, Microsoft ha hecho de modelado entidad-relación se pueda usar programando. Lo consigue mediante una combinación de archivos de esquema XML, detrás está el código .NET y el API de ADO.NET Entity Framework. Los archivos de esquema definen una capa conceptual donde muestran el almacén de datos y crean un mapa entre los datos y los objetos de nivel de aplicación. ADO.NET Entity Framework permite escribir sus programas usando las clases que se generan a partir de esta capa conceptual.[50]

Arquitectura de Entity Framework

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.[51]

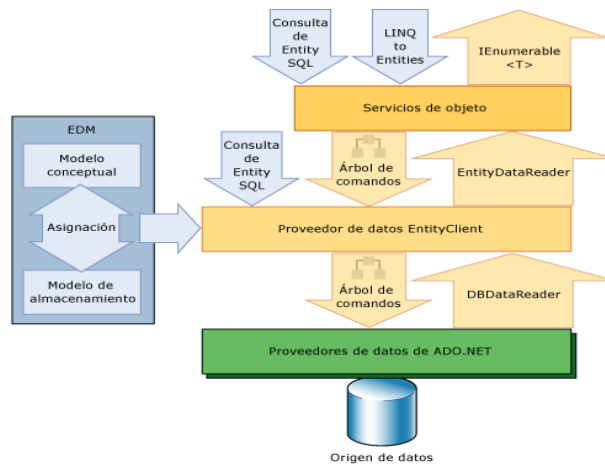


Figura 4.5: Arquitectura y componentes Entity Framework.
 Fuente: “Microsoft”,
[https://msdn.microsoft.com/es-es/library/bb399567\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/bb399567(v=vs.110).aspx)

Proveedores de Datos en Entity Framework

Por defecto ADO.NET Entity Framework sólo está disponible de base para SQL Server. Pero también .NET Framework incluye proveedores para otras bases de datos con los controladores ODBC y OLE DB. Para acceder directamente a otras bases de datos, muchos proveedores de terceros están disponibles como lo son, MySQL Connector / NET y Oracle Data Provider para .NET. [52]

■ Ventajas

- ADO.NET Entity Framework permite a los desarrolladores crear aplicaciones de acceso a datos, programando con un modelo de aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y el mantenimiento necesarios para las aplicaciones orientadas a datos.
- Las aplicaciones pueden funcionar en términos de un modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones.
- Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento.
- Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación.

- La compatibilidad con Language Integrated Query (LINQ) proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual. [53]
- EDM soluciona los desafíos que plantea el tener datos almacenados en muchos formatos, describiendo la estructura de los datos en forma de entidades y relaciones, que son independientes de cualquier esquema de almacenamiento. Esto hace que el formato en el que están almacenados los datos sea irrelevante a la hora de diseñar y desarrollar las aplicaciones. Dado que las entidades y las relaciones describen la estructura de los datos tal como se usan en una aplicación (no el formato en el que están almacenados), pueden evolucionar al mismo tiempo que la aplicación.[54]

■ **Desventajas**

- Uno de los principales problemas que se podría topar cuando se trata de desarrollos para Aplicaciones Empresariales, donde la base de datos sea muy grande, es el rendimiento.

4.4.4.4. Cuadro Comparativo Tecnologías de Acceso a Datos

Característica	ADO.NET	LINQ to SQL	Entity Framework
Proveedores	SQL para Microsoft SQL Server y el proveedor OLE DB para cualquier origen de datos compatible con OLE DB.	LINQ to SQL sólo permite conectarse a SQL Server	Por defecto disponible para SQL Server, pero también incluye proveedores para otras bases de datos como MySql y Oracle.
Compatibilidad de Language Integrated Query (LINQ)	No	Si	Si
Permite el mapeo objeto-relacional (ORM)	No	Si	Si
Ahorro de código y facilidad para realizar consultas	Bajo	Medio	Alto

Tabla 4.51: Cuadro Comparativo Tecnologías de Acceso a Datos

Característica	ADO.NET	LINQ to SQL	Entity Framework
Permite el Modelado de datos conceptual (EDM)	No	No	Si
Permite la herencia, entidades lógicas o entidades de negocio	No	No	Si
Gestión de Base de datos	Si	Si	Si
Gestión de entidades y relaciones	No	No	Si
Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento	No	No	Si
Puede generar la base de datos del modelo	No	No	Si
Se puede utilizar para el desarrollo rápido, flexible y fácil en el acceso a datos	No	Si	Si
Tiene asistentes que permiten modelar datos	No	No	Si
Compatible con el modelo de características como la de relaciones de muchos-a-mucho	No	No	Si

Tabla 4.52: Cuadro Comparativo Tecnologías de Acceso a Datos

Análisis del Cuadro Comparativo

Analizando el cuadro anterior se puede ver las grandes características que posee Entity Framework, ya que a diferencia de ADO.NET permite una nueva forma de diseño de los datos, debido a que trabaja con el modelado de datos, a la vez Entity Framework, permite un desarrollo mucho más rápido de la aplicación a desarrollar. Entity Framework (EF) permite una gran velocidad de desarrollo a diferencia de ADO.NET.

En ADO.NET hay que hacer todo manualmente, por otra parte EF permite a los desarrolladores crear aplicaciones de acceso a datos programando con un modelo de

aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional.

Así también Entity Framework lleva ventaja sobre LINQ to SQL. LINQ to SQL sólo permite conectarse a SQL Server y EF en cambio soporta varios proveedores como MySQL, Oracle, SQL Server, XML, etc.

Otras ventajas de EF sobre LINQ to SQL es que LINQ to SQL mapea cada tabla/vista como una clase o entidad sin permitir herencia, entidades lógicas o entidades de negocio, en cambio EF lo implementa completamente.

Entity Framework gestiona las relaciones n-n (muchos-a-muchos) y LINQ to SQL no gestiona relaciones n-n (muchos-a-muchos).

En vista de que Entity Framework brinda grandes características y muchas más funcionalidades que las otras dos tecnologías de acceso a datos, Entity Framework es la tecnología elegida para el acceso a datos del sistema de gestión de servicios comunitarios del DIBESAU de la UTA, ya que lo que se busca es una nueva forma de modelar datos con el objetivo de reducir el tiempo de desarrollo, ahorro de código y facilidad para realizar consultas.

Así también al utilizar el motor de base de datos SQL-Server el proveedor de Entity Framework viene por defecto lo cual ayudará de una manera mas fácil a la conexión de la base de datos. Por otra parte no se cuenta con gran cantidad de tablas, por lo que el rendimiento del sistema no se verá afectado.

Una vez elegida como tecnología de acceso a datos a Entity Framework, se procede a continuación a detallar la misma.

4.4.4.5. Entity Framework (EF)

ADO.NET Entity Framework admite aplicaciones y servicios centrados en datos, y proporciona una plataforma para la programación con datos que eleva el nivel de abstracción del nivel lógico relacional al nivel conceptual. Al permitir a los programadores trabajar con datos en un nivel de abstracción superior, Entity Framework admite código que es independiente de cualquier motor de almacenamiento de datos o esquema relacional determinados.

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. El modelo conceptual, el modelo de almacenamiento y las asignaciones entre los dos se expresan en esquemas basados en XML.[51]

Especificación de los Componentes de Entity Framework

ADO.NET Entity Framework se apoya en seis elementos fundamentales construidos encima de toda las bases de ADO.NET, tal y como se puede ver en la siguiente figura.

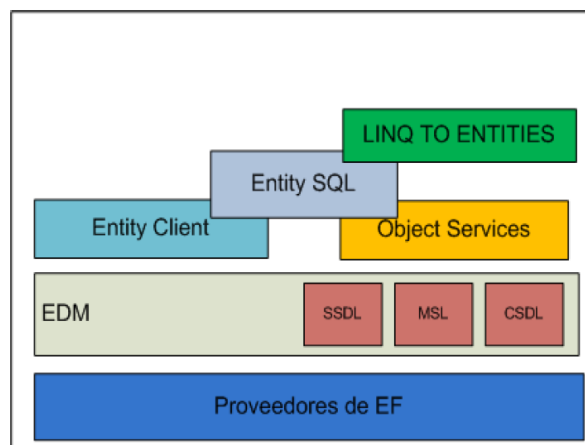


Figura 4.6: Arquitectura y componentes Entity Framework.
Fuente: Unai Zorrilla Castro. (2011), ADO.NET Entity Framework 4.1
Aplicaciones y servicios centrados en datos.

A continuación, se presenta los conceptos fundamentales relacionados con cada uno de estos elementos (de abajo hacia arriba).

Proveedores de EF

Como ya se mencionó anteriormente, por defecto ADO.NET Entity Framework sólo está disponible de base para SQL Server. Pero también .NET Framework incluye proveedores para otras bases de datos con los controladores ODBC y OLE DB. Para acceder directamente a otras bases de datos, muchos proveedores de terceros están disponibles como lo son, MySQL Connector / NET y Oracle Data Provider para .NET.[52]

Entity Data Model (EDM)

El primero de los elementos fundamentales en la arquitectura de ADO.NET Entity Framework es el Modelo de Entidades, normalmente conocido como EDM.

El Entity Data Model (EDM) es un modelo de entidad relación. El EDM define los datos en un formato neutro, que no está restringido por la estructura de los lenguajes de programación o las bases de datos relacionales. Los esquemas EDM se usan para especificar los detalles de las entidades y las relaciones y para implementarlos como estructuras de datos.

■ Entidad

- Es una parte del dominio de una aplicación que se debe representar mediante datos, las entidades son las especificaciones formales de todos los detalles de la estructura de datos.
- Un tipo de datos Customer, por ejemplo, se diseña para contener detalles como el nombre de contacto, nombre de compañía, dirección e identificador único para cada cliente. Un tipo Order podría contener la fecha del pedido, las líneas de pedido, la fecha de vencimiento y un identificador único, entre otros datos. Igual que en la mayor parte de los demás modelos de datos.
- El EntityType del EDM es la especificación de un tipo de datos que representa la entidad en el dominio de la aplicación.

■ Relación

- Es la conexión lógica entre entidades: por ejemplo, la conexión lógica entre el pedido de una mercancía y el cliente que lo realiza. Dado que un cliente puede tener asociados muchos pedidos, la relación entre un cliente y sus pedidos es de uno a varios. Los productos y los suministradores podrían tener una relación varios a varios.
- AssociationType representa una asociación en el Entity Data Model (EDM).[55]

Herramientas de Entity DataModel

■ Diseñador de modelos EDM (Entity Data Model Designer)

El diseñador de modelos es una herramienta visual integrada dentro de Visual Studio 2010 que permite crear y editar modelos conceptuales. Este diseñador está formado por los siguientes componentes:

- Una superficie de diseño para crear y editar los modelos de una forma rápida similar al trabajo con los diagramas de clases de Visual Studio.

- Una ventana de Detalles de mapeo, que permitirá ver y editar los mapeos entre los elementos del modelo conceptual y del esquema de la base de datos con la que se está trabajando.
- Una ventana de Navegación por el modelo, la cual permitirá ver árboles de información sobre el modelo conceptual y el modelo físico.
- Nuevos elementos dentro de la ventana de herramientas, que permitirán, por ejemplo, crear las entidades, las asociaciones o relaciones de herencia.

El diseñador de modelos opera sobre ficheros de tipo edmx. Estos ficheros se forman mediante la combinación de tres secciones de metadatos en formato XML (que en ocasiones pueden presentarse también como ficheros independientes), llamadas respectivamente SSDL, CSDL y MSL.[56]

- **Lenguaje de definición de esquemas de almacenamiento (SSDL)**
 - El lenguaje de definición de esquemas de almacenamiento (SSDL) define el modelo de almacenamiento, que también se denomina modelo lógico. La extensión de archivo es .ssdl.
 - Describe la estructura física de la base de datos subyacente, incluyendo la definición de las tablas, columnas, vistas, procedimientos almacenados y relaciones entre los distintos objetos de la misma.
- **Lenguaje de definición de esquemas conceptuales (CSDL)**
 - El lenguaje de definición de esquemas conceptuales (CSDL) define el modelo conceptual. CSDL es la implementación de Entity Framework del Entity Data Model. La extensión de archivo es .csdl.
 - Describe las entidades que se desean tener en el modelo conceptual, así como las propiedades de navegación o asociaciones entre las distintas entidades.
- **Lenguaje de especificación de asignaciones (MSL)**
 - El lenguaje de especificación de asignaciones (MSL) define las asignaciones entre los modelos conceptual y de almacenamiento. La extensión de archivo es .msl. [51]

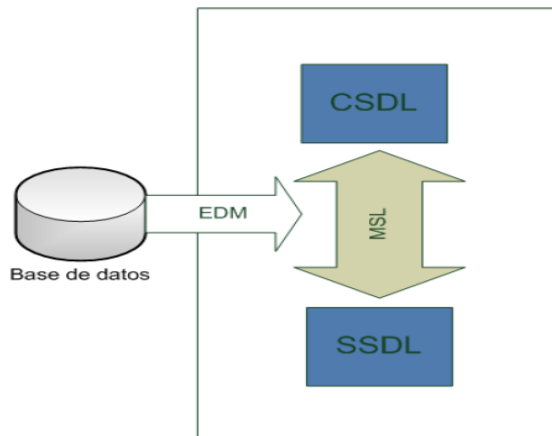


Figura 4.7: Secciones de metadatos de un modelo EDMX.
 Fuente: “Microsoft”,
[https://msdn.microsoft.com/es-es/library/bb399567\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/bb399567(v=vs.110).aspx)

Entity Framework usa un modelo conceptual para proporcionar una vista de los datos centrada en objetos, expresados como tipos de entidad y asociaciones. Los desarrolladores de aplicaciones solo tienen que dedicarse a programar con las clases generadas a partir del modelo conceptual, en lugar de tener que pensar también en el esquema de almacenamiento y en la forma de obtener acceso a los objetos del almacén de datos y transformarlos en objetos de programación.

Entity Framework usa un modelo conceptual, un modelo de almacenamiento y la asignación entre ambos modelos para transformar las operaciones de creación, lectura, actualización y eliminación de las entidades en las operaciones equivalentes en el origen de datos.[57]

- **Asistente de modelos de entidades (Entity Data Model Wizard)**

- El Asistente para Modelo de datos de entidad genera un archivo .edmx. Puede crear un modelo conceptual, un modelo de almacenamiento y las asignaciones a partir de una base de datos existente. O bien, puede crear un archivo .edmx vacío y utilizar el Diseñador Entity Data Model para crear un modelo conceptual.[58]

- **Asistente para Actualizar Modelo**

- Esta herramienta se utiliza para actualizar el modelo EDM después de que se hayan realizado cambios en la base de datos con la que se

está trabajando. Para usar esta herramienta, basta con situarse en el Navegador del modelo y seleccionar de su menú contextual la opción “Actualizar modelo desde la base de datos”.[56]

■ **Asistente para Generar Base de datos**

- El Asistente para generar base de datos genera un modelo de almacenamiento, las asignaciones y el lenguaje de definición de datos (DDL) para crear una base de datos a partir de un modelo conceptual existente. El asistente permite hacer primero el modelado conceptual y después crear una base de datos que admite el modelo.[58]

Entity Client

El proveedor de EntityClient es un proveedor de datos que usan las aplicaciones de Entity Framework para tener acceso a los datos descritos en un Entity Data Model (EDM). EntityClient utiliza otros proveedores de datos de .NET Framework para tener acceso al origen de datos. Por ejemplo, EntityClient utiliza el Proveedor de datos de .NET Framework para SQL Server (SqlClient) al tener acceso a una base de datos de SQL Server.[59]

La diferencia principal de Entity Client con el resto de proveedores radica en que, a diferencia de los proveedores tradicionales que trabajan con los modelos físicos de sus bases de datos, con Entity Client se consulta los modelos de EDM.

Entity Client implementa también a su nivel una arquitectura abierta, y es capaz de trabajar con diferentes proveedores de EF específicos, que son quienes se encargan de traducir las consultas sobre el modelo en consultas en el dialecto específico de la base de datos subyacente, así como de la ejecución de dichas consultas y la recuperación de los resultados.[56]

El lenguaje que se utiliza para consultar los modelos de EDM se llama Entity SQL (abreviadamente, eSQL).

■ **Entity SQL (eSQL)**

- Es una variante de los dialectos de SQL.
- Mejora ciertos aspectos de las consultas, como la navegación entre tablas.

Con Entity Client y eSQL realmente se puede cubrir la mayoría de las necesidades de una capa de acceso a datos.

- Se puede consultar modelos conceptuales de forma similar a cómo se consultaban BD en ADO.NET

- Dispone de clases equivalentes a las ya conocidas.

Sin embargo, todavía se tiene que realizar una transformación de los datos recuperados a objetos del dominio (materialización). Para evitar este paso, EF ofrece una nueva capa: Object Services. [56]

Object Services

Object Services es un componente de Entity Framework que permite consultar, insertar, actualizar y eliminar datos.

Object Services admite tanto consultas LINQ (Language-Integrated Query) como consultas Entity SQL con los tipos que se definen en un modelo Entity Data Model (EDM).

Object Services materializa los datos devueltos como objetos y propaga los cambios de los objetos de vuelta al origen de datos. [60]

Conjunto de clases que permiten consultar y obtener resultados en términos de objetos y por ende se reduce la cantidad y la complejidad del código.

- **LINQ To Entities (L2E)**
 - EF cuenta con la compatibilidad con Language Integrated Query (LINQ).
 - La compatibilidad con Language Integrated Query (LINQ) proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.
 - LINQ que hace posible las consultas integradas sobre los modelos de entidades se llama precisamente LINQ to Entities (L2E), y ha sido construido sobre la base de Object Services.
 - LINQ to Entities proporciona la capacidad de realizar consultas integradas en lenguajes (LINQ) que permite a los desarrolladores de software escribir consultas contra el modelo conceptual de Entity Framework.
 - Las consultas con Entity Framework se representan mediante consultas de árboles de comandos. Un árbol de comandos es una representación de una consulta que es compatible con EF, que se ejecutan en el contexto del objeto.
 - LINQ to Entities convierte las consultas de Language-Integrated Queries (LINQ) en consultas de árboles de comandos, ejecuta las consultas en Entity Framework y devuelve objetos que se pueden usar tanto en Entity Framework como en LINQ. [56]

Cuadro de los Operadores de Consulta para LINQ To Entities (L2E) y Entity SQL (eSQL)

Operadores de Consulta	Definición	LINQ To Entities (L2E)	Entity SQL (eSQL)
Proyección y filtrado	La proyección hace referencia a la transformación de los elementos de un conjunto de resultados en un formato deseado.	Los métodos de proyección son: <ul style="list-style-type: none"> ▪ Select ▪ SelectMany El método de filtrado es: <ul style="list-style-type: none"> ▪ Where 	El método de proyección es: <ul style="list-style-type: none"> ▪ SELECT ▪ SELECT VALUE El método de filtrado es: <ul style="list-style-type: none"> ▪ WHERE
Combinación	Una combinación de dos orígenes de datos es la asociación de objetos en un origen de datos con objetos de otro origen de datos que comparten un atributo o propiedad comunes.	<ul style="list-style-type: none"> ▪ Join ▪ GroupJoin 	<ul style="list-style-type: none"> ▪ JOIN
Agregación	Una operación de agregación calcula un valor único a partir de una colección de valores.	<ul style="list-style-type: none"> ▪ Average ▪ Count ▪ Max ▪ Min ▪ Sum 	<ul style="list-style-type: none"> ▪ AVG ▪ MAX ▪ MIN ▪ SUM

Tabla 4.53: Cuadro de los Operadores de Consulta para LINQ To Entities (L2E) y Entity SQL (eSQL).

Operadores de Consulta	Definición	LINQ To Entities (L2E)	Entity SQL (eSQL)
Operadores de Conjunto	son operaciones de consulta que basan sus conjuntos de resultados en la presencia o ausencia de elementos.	<ul style="list-style-type: none"> ▪ All ▪ Any ▪ Concat ▪ Contains ▪ Distinct ▪ EqualAll ▪ Except ▪ Intersect ▪ Union 	<ul style="list-style-type: none"> ▪ ANYELEMENT ▪ DISTINCT ▪ EXCEPT ▪ INTERSECT ▪ UNION ▪ [NOT] EXISTS ▪ [NOT] IN
Agrupamiento	El agrupamiento hace referencia a la operación de colocar los datos en grupos de manera que los elementos de cada grupo compartan un atributo común	<ul style="list-style-type: none"> ▪ GroupBy 	<ul style="list-style-type: none"> ▪ GROUP BY
Operadores de Tipo	Se encargan de la conversión y prueba de tipos se admiten en el contexto de Entity Framework.	<ul style="list-style-type: none"> ▪ Convert ▪ OfType 	<ul style="list-style-type: none"> ▪ OFTYPE

Tabla 4.54: Cuadro de los Operadores de Consulta para LINQ To Entities (L2E) y Entity SQL (eSQL).

Formas de trabajo con Entity Framework

▪ Database First

- Esta forma es cuando ya se posee una base de datos o si se desea utilizar otra herramienta para crearla. EF toma la base de datos ya creada y realiza ingeniería inversa para crear las clases (entidades) y mapeos.

- Este modelo, mapea todas las tablas, vistas, procedimientos almacenados y columnas, en clases, métodos y propiedades.
- **Model First**
 - Esta forma sirve cuando no se posee una base de datos y se desea trabajar en un ambiente visual para crear el modelo y la base de datos.
 - Una vez terminado, se puede generar código para crear la base de datos, como también las entidades o clases que se usarán en el código.
- **Code First**
 - Este modelo sirve cuando se está en total control de todos los aspectos del ciclo de vida del desarrollo.
 - Aquí se definen, de manera manual, las clases, relaciones y mapeos, después se genera código para crear la base de datos.[47]

4.5. Fase de Diseño

4.5.1. Diagramas de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. [61]

Diseño de Diagramas de Secuencia

A través de los diagramas de secuencia se puede entender gráficamente el comportamiento y comunicación de los objetos dentro de la aplicación así como las acciones para cumplir las tareas definidas en cada caso de uso.

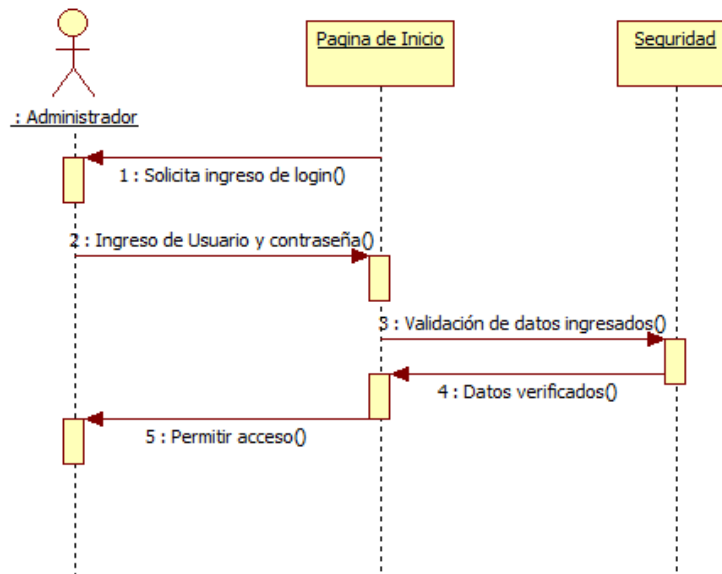


Figura 4.8: Diagrama de Secuencia Iniciar sesión
Fuente: StartUML

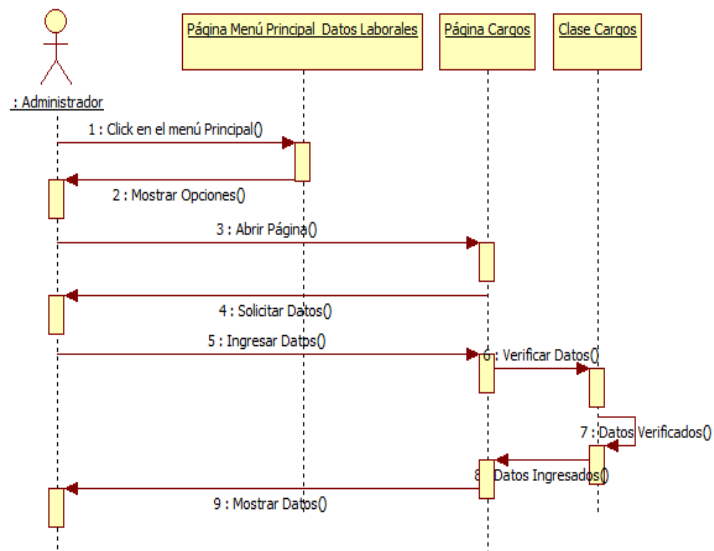


Figura 4.9: Diagrama de Secuencia Ingresar Cargos-Responsables
Fuente: StartUML

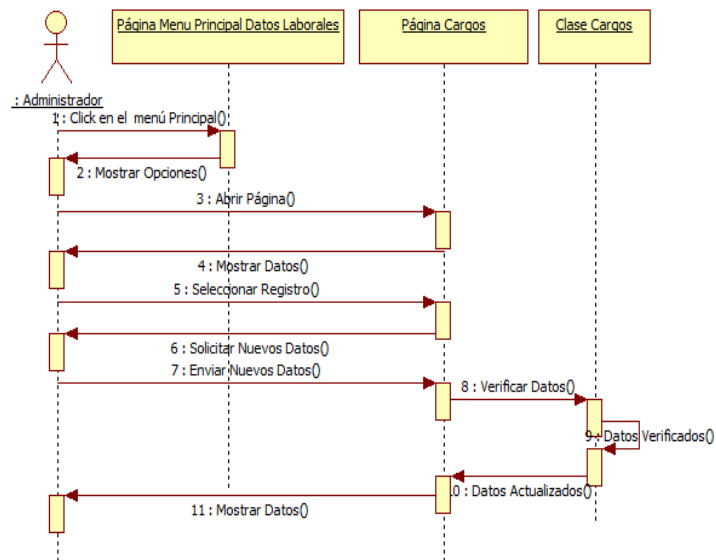


Figura 4.10: Diagrama de Secuencia Actualizar Cargos-Responsables
Fuente: StartUML

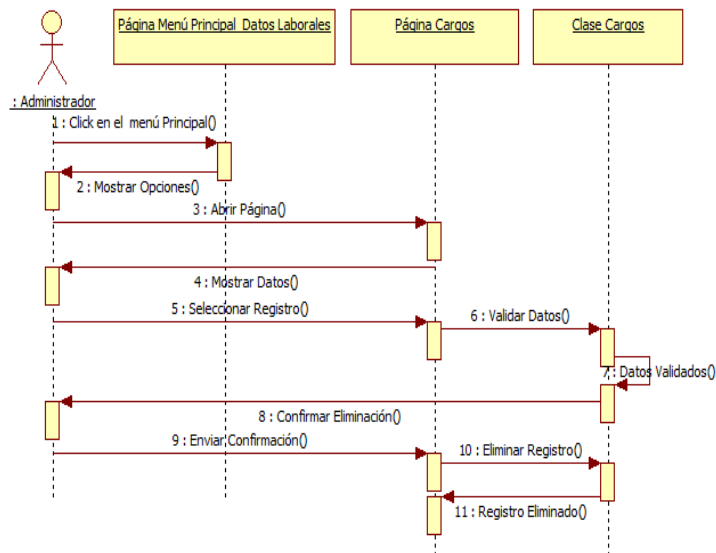


Figura 4.11: Diagrama de Secuencia Eliminar Cargos-Responsables
Fuente: StartUML

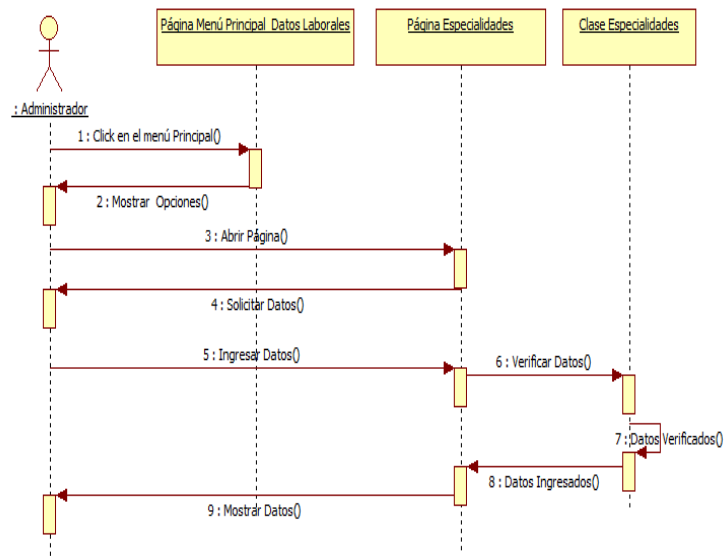


Figura 4.12: Diagrama de Secuencia Ingresar Especialidades-Responsables
Fuente: StartUML

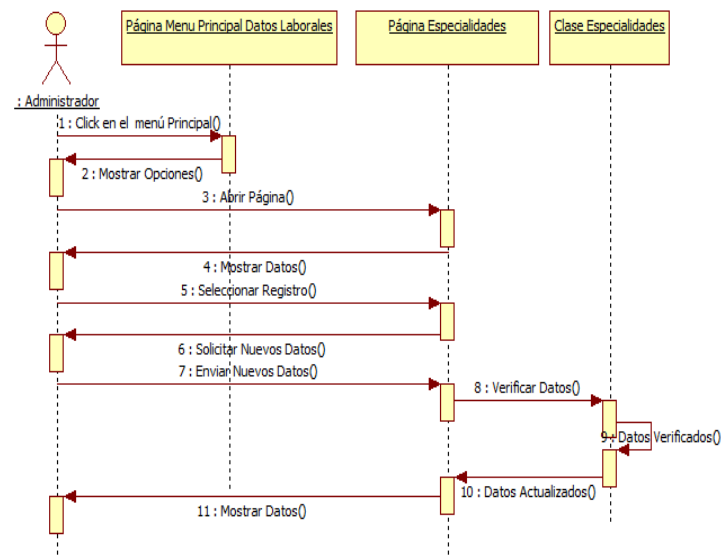


Figura 4.13: Diagrama de Secuencia Actualizar Especialidades-Responsables
Fuente: StartUML

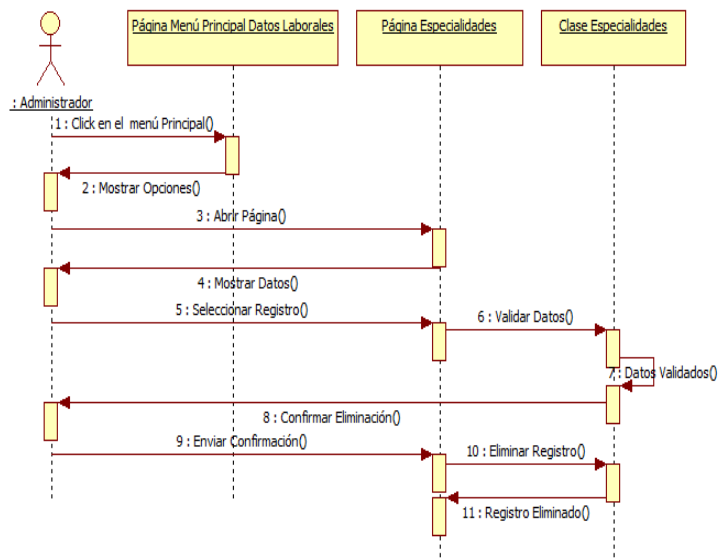


Figura 4.14: Diagrama de Secuencia Eliminar Especialidades-Responsables
Fuente: StartUML

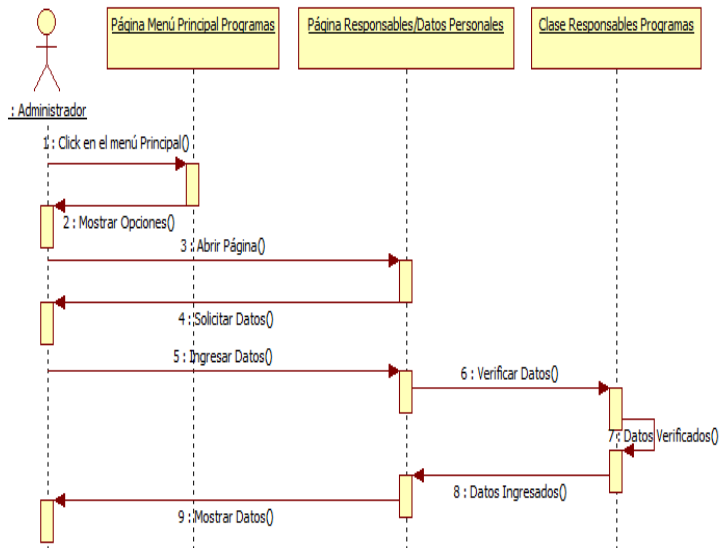


Figura 4.15: Diagrama de Secuencia Ingresar Responsables-Programas
Fuente: StartUML

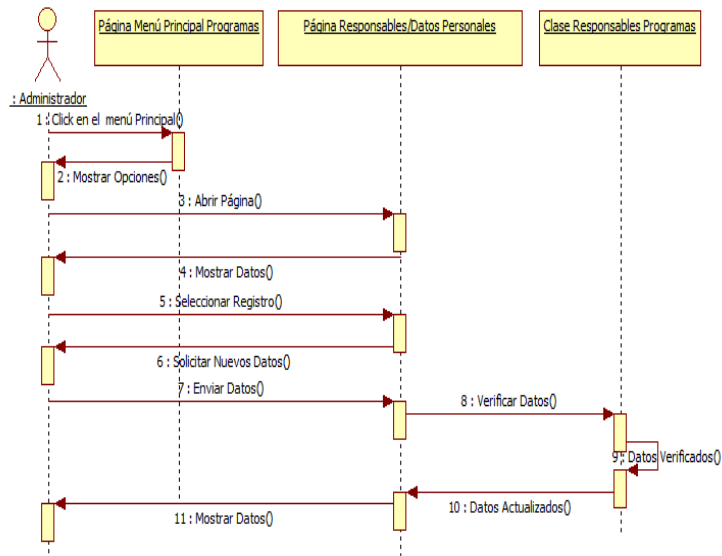


Figura 4.16: Diagrama de Secuencia Actualizar Responsables-Programas
Fuente: StartUML

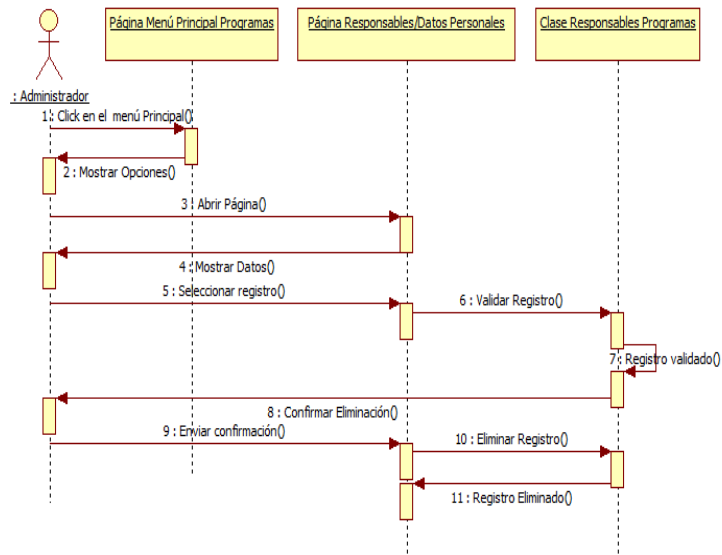


Figura 4.17: Diagrama de Secuencia Eliminar Responsables-Programas
Fuente: StartUML

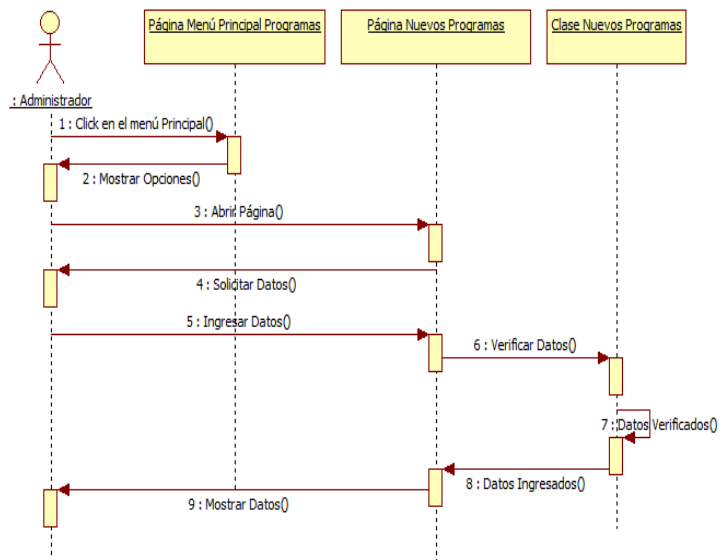


Figura 4.18: Diagrama de Secuencia Ingresar Programas
Fuente: StartUML

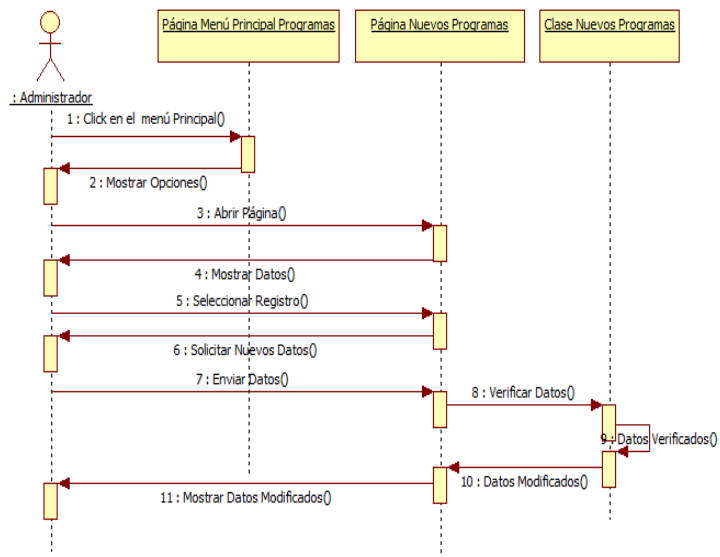


Figura 4.19: Diagrama de Secuencia Actualizar Programas
Fuente: StartUML

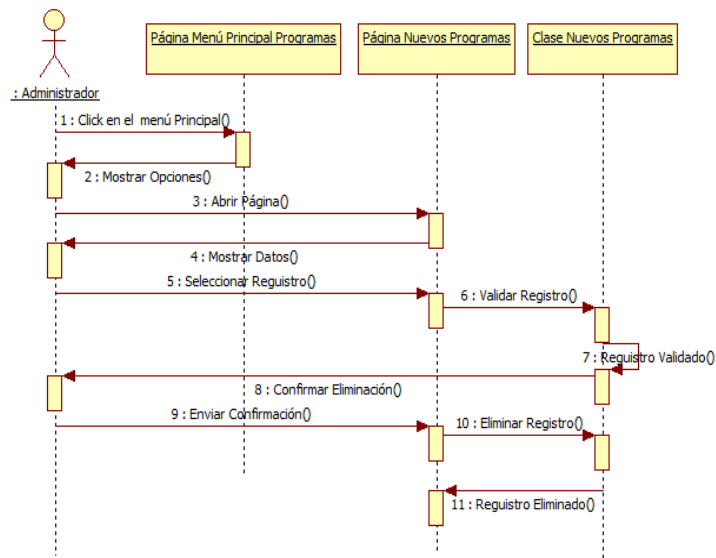


Figura 4.20: Diagrama de Secuencia Eliminar Programas
Fuente: StartUML

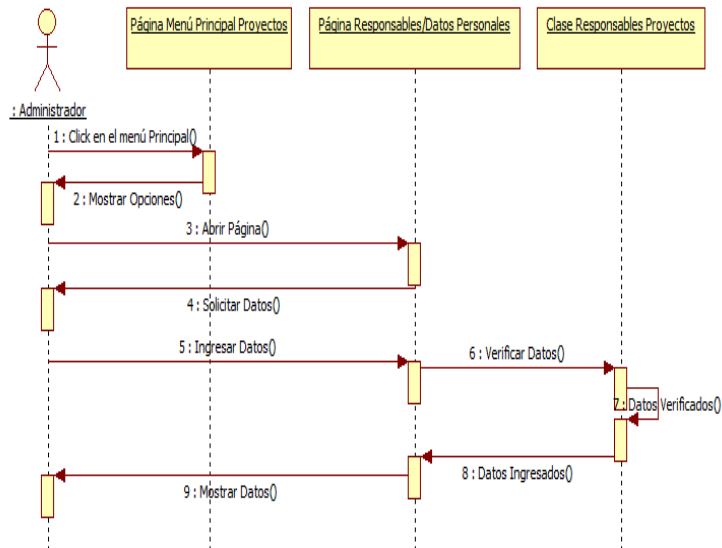


Figura 4.21: Diagrama de Secuencia Ingresar Responsables-Proyectos
Fuente: StartUML

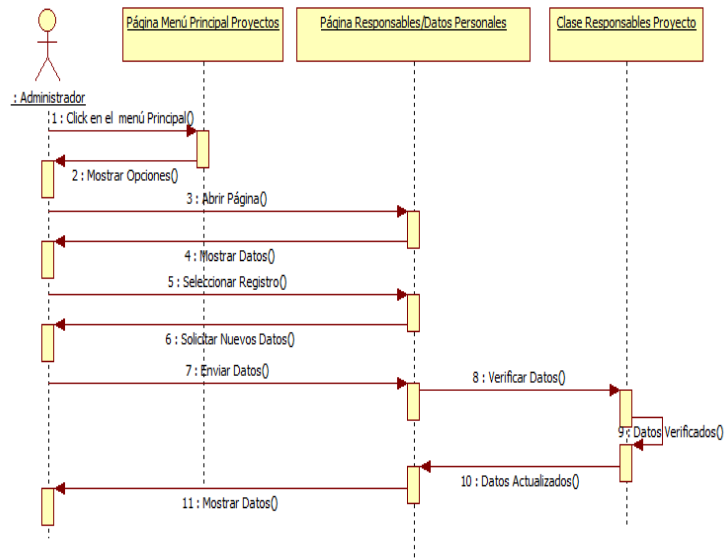


Figura 4.22: Diagrama de Secuencia Actualizar Responsables-Proyectos
Fuente: StartUML

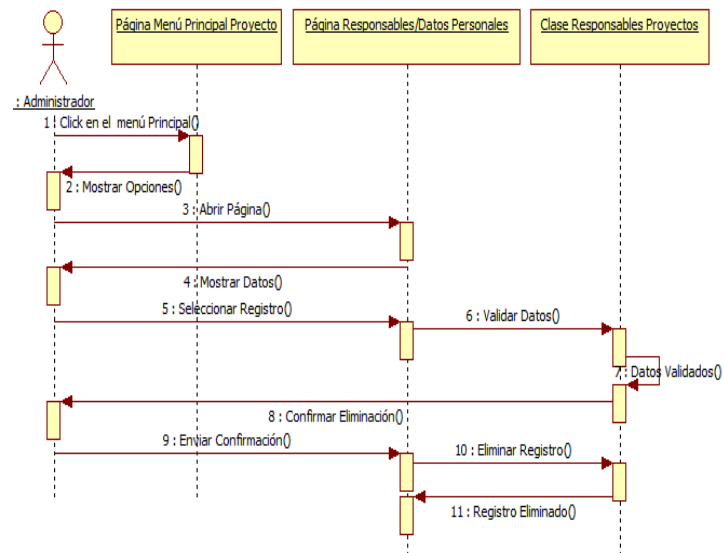


Figura 4.23: Diagrama de Secuencia Eliminar Responsables-Proyectos
Fuente: StartUML

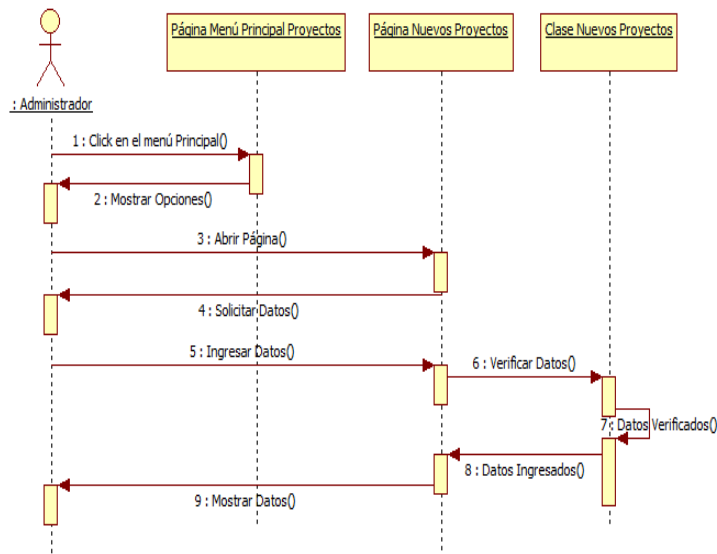


Figura 4.24: Diagrama de Secuencia Ingresar Proyectos
Fuente: StartUML

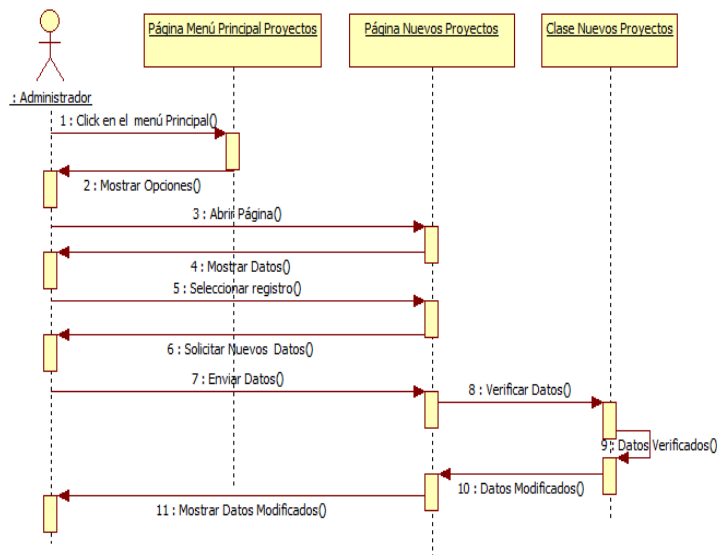


Figura 4.25: Diagrama de Secuencia Actualizar Proyectos
Fuente: StartUML

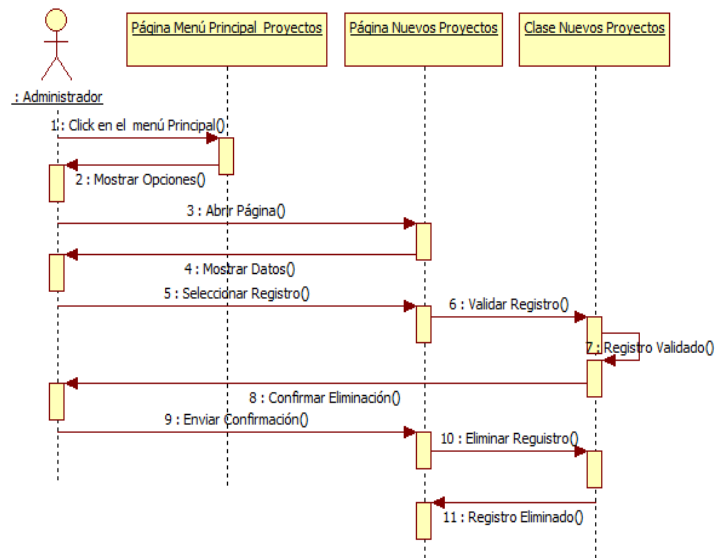


Figura 4.26: Diagrama de Secuencia Eliminar Proyectos
Fuente: StartUML

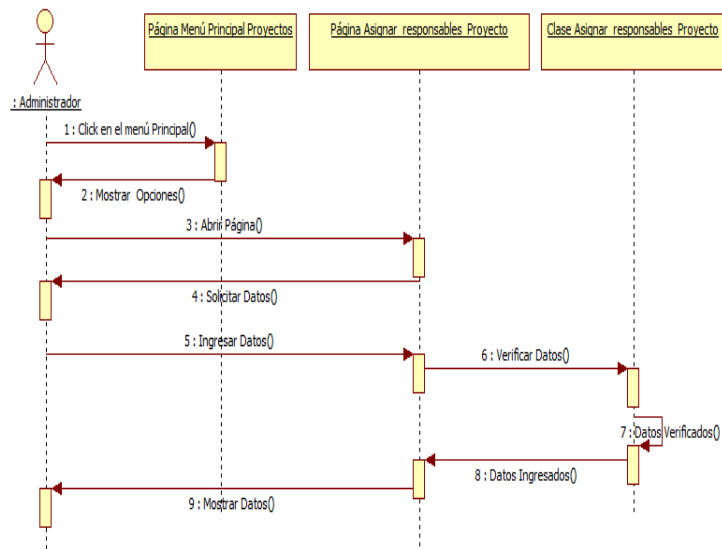


Figura 4.27: Diagrama de Secuencia Ingresar Asignación de responsables a Proyectos
Fuente: StartUML

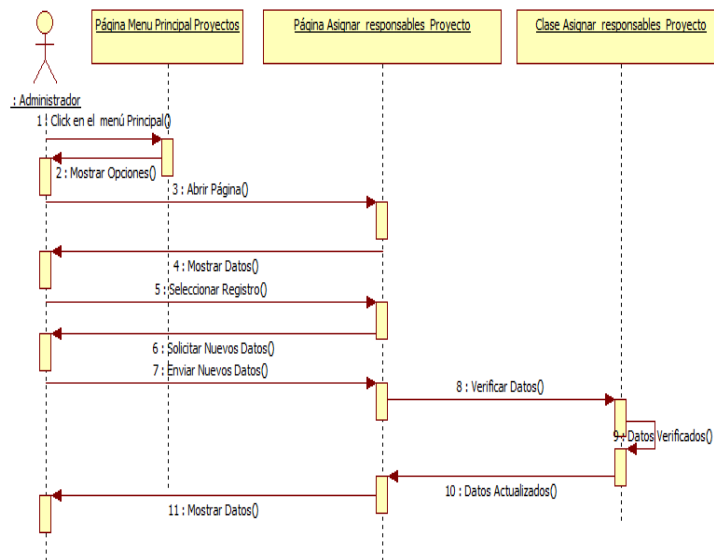


Figura 4.28: Diagrama de Secuencia Actualizar Asignación de responsables a Proyectos

Fuente: StartUML

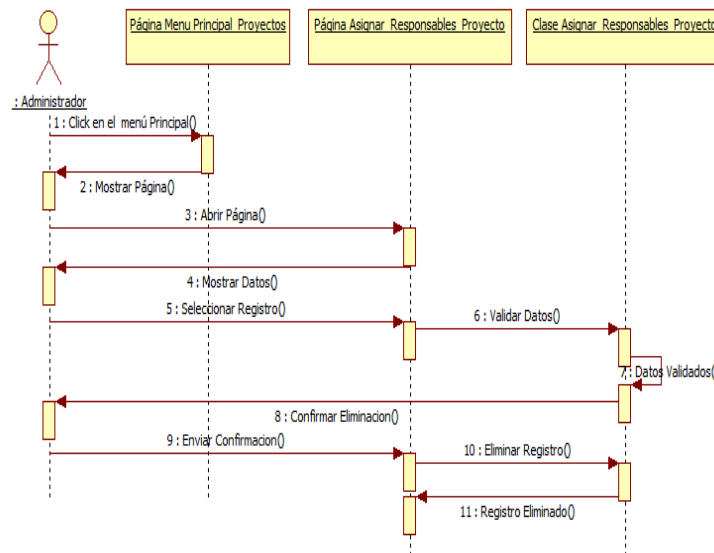


Figura 4.29: Diagrama de Secuencia Eliminar Asignación de responsables a Proyectos

Fuente: StartUML

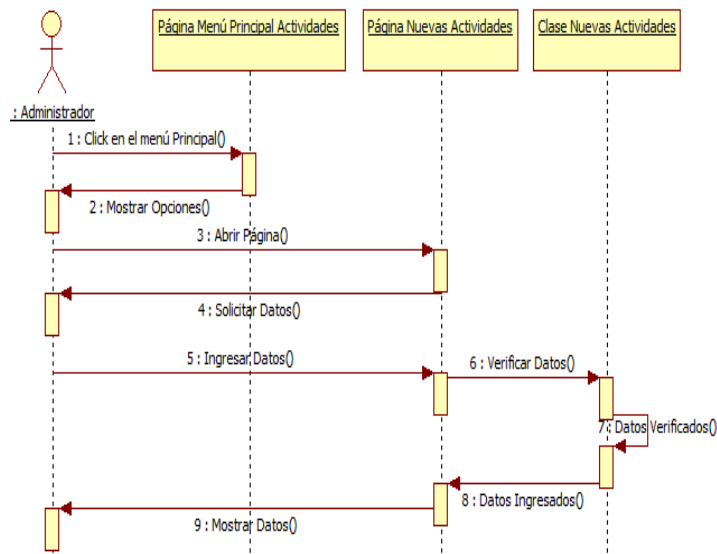


Figura 4.30: Diagrama de Secuencia Ingresar Actividades
Fuente: StartUML

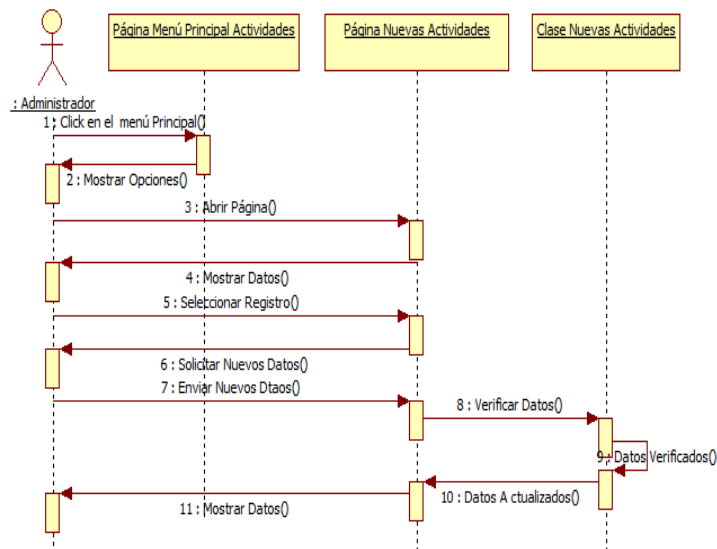


Figura 4.31: Diagrama de Secuencia Actualizar Actividades
Fuente: StartUML

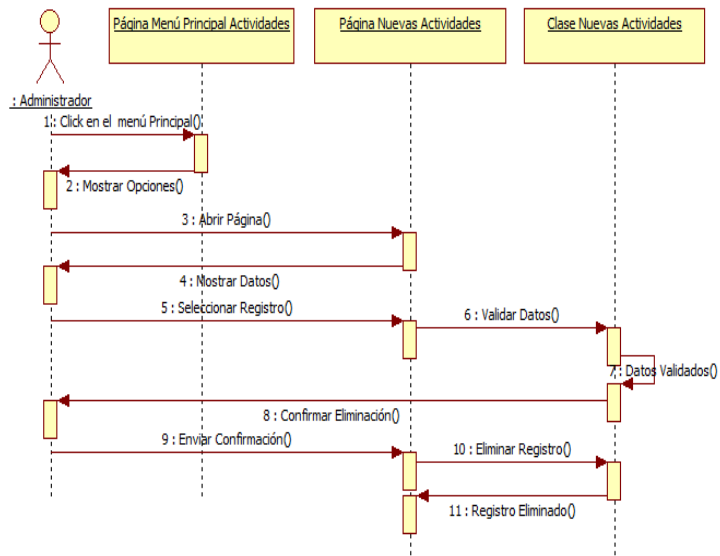


Figura 4.32: Diagrama de Secuencia Eliminar Actividades
Fuente: StartUML

4.5.2. Diseño del Modelo de Base de Datos

A continuación se muestran las tablas y relaciones de la base de datos denominada `dbe_dibesau`.

En cada una de las tablas se puede observar su respectivo nombre y el tipo de dato que tiene cada uno de sus campos.

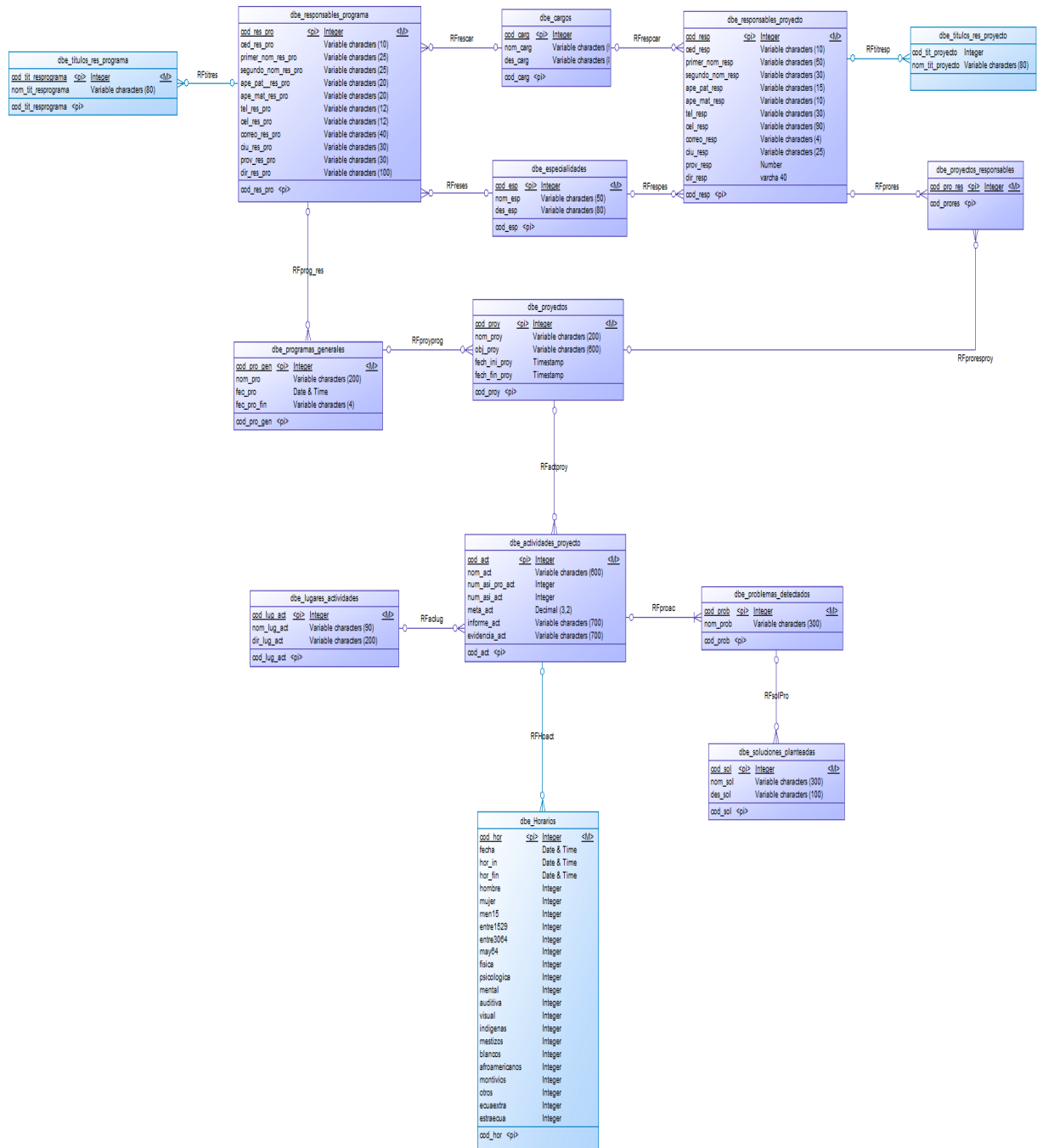


Figura 4.33: Entity Data Model (EDM) db_e_DUBESAU
Elaborado por: Amalia Núñez

Diseño Lógico de la Base de Datos

A continuación se muestran las relaciones correspondientes de cada una de las tablas con sus respectivas claves foráneas.

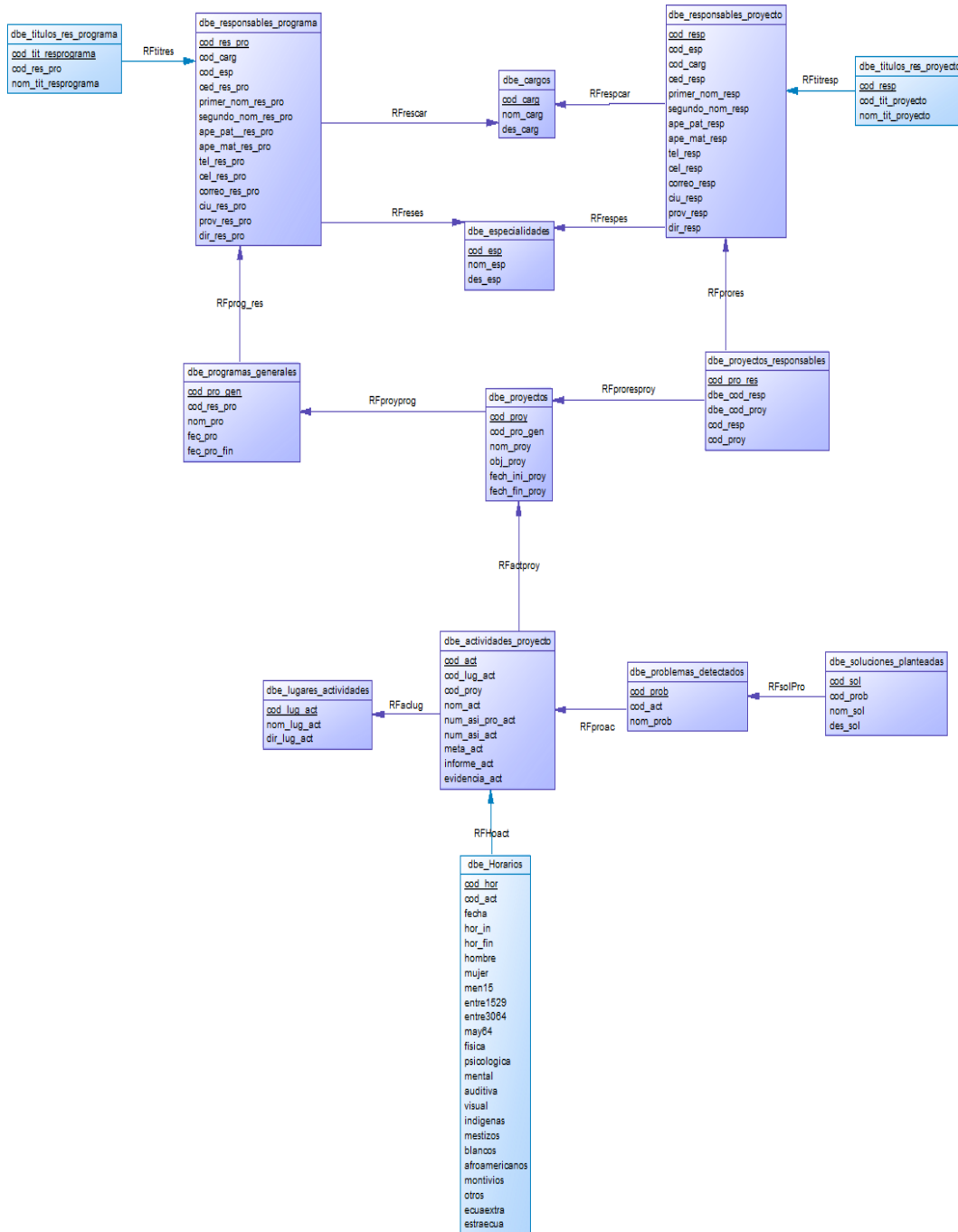


Figura 4.34: Entity Data Model (EDM) db_e_DUBESAU
Elaborado por: Amalia Núñez

Modelo de Entidad de Datos o Entity Data Model (EDM - Modelo de Clases)

A continuación se muestra un mapeo objeto relacional, a partir de este modelo se generan los objetos de negocio y la Data Access Layer (DAL) o Capa de Acceso a Datos.

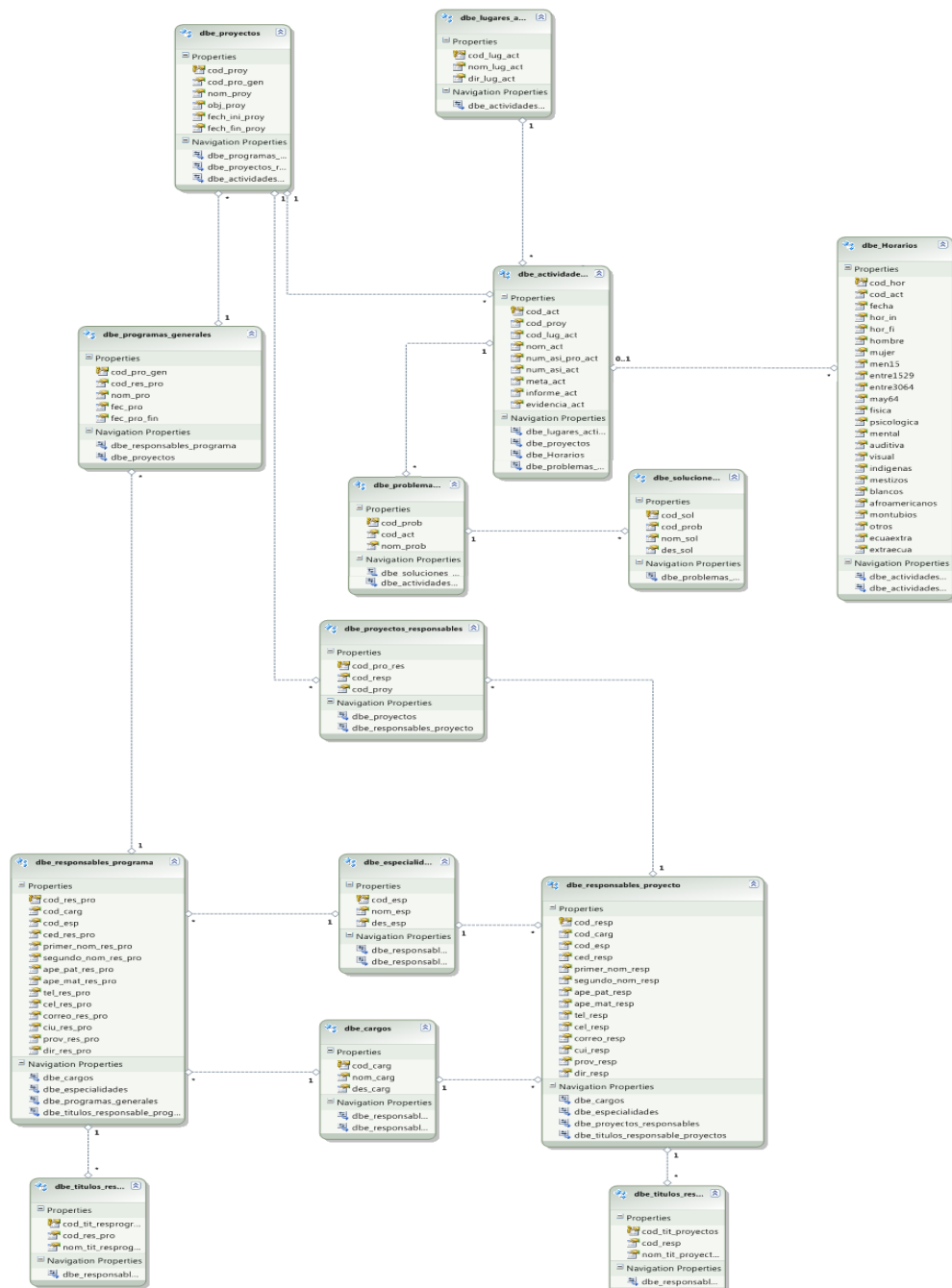


Figura 4.35: Entity Data Model (EDM) db_e_DUBESAU
Elaborado por: Amalia Núñez

4.5.3. Realización del Diccionario de Datos

Tabla Cargos

dbe_cargos	Tabla Cargos			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_carg	Código	int		entero largo
nom_carg	Nombre	varchar	50	texto
des_carg	Descripción	varchar	80	texto

Tabla 4.55: Diccionario de Datos Tabla Cargos

Elaborado por: Amalia Núñez

dbe_cargos	Tabla Cargos			
Nombre	REQ	PK	FK	Tabla Padre
cod_carg	si	si	no	
nom_carg	si	no	no	
des_carg	no	no	no	

Tabla 4.56: Diccionario de Datos Tabla Cargos

Elaborado por: Amalia Núñez

Tabla Especialidades

dbe_especialidades	Tabla Especialidades			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_esp	Código	int		entero largo
nom_esp	Nombre	varchar	50	texto
des_esp	Descripción	varchar	80	texto

Tabla 4.57: Diccionario de Datos Tabla Especialidades

Elaborado por: Amalia Núñez

dbe_espacialidades	Tabla Cargos			
Nombre	REQ	PK	FK	Tabla Padre
cod_esp	si	si	no	
nom_esp	si	no	no	
des_esp	no	no	no	

Tabla 4.58: Diccionario de Datos Tabla Especialidades

Elaborado por: Amalia Núñez

Tabla Responsables Programa

dbe_responsables_programa	Tabla Responsables Programa			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_res_pro	Código	int		entero largo
cod_carg	Código	int		entero largo
cod_esp	Código	int		entero largo
ced_res_pro	Cédula	varchar	10	texto
primer_nom_res_pro	Primer nombre	varchar	25	texto
segundo_nom_res_pro	Segundo Nombre	varchar	25	texto
ape_pat_res_pro	Apellido Paterno	varchar	20	texto
ape_mat_res_pro	Apellido Materno	varchar	20	texto
tel_res_pro	Teléfono	varchar	12	texto
cel_res_pro	Celular	varchar	12	texto
correo_res_pro	Correo	varchar	40	texto
ciu_res_pro	Ciudad	varchar	30	texto
prov_res_pro	Provincia	varchar	30	texto
dir_res_pro	Dirección	varchar	100	texto

Tabla 4.59: Diccionario de Datos Tabla Responsables Programa

Elaborado por: Amalia Núñez

dbe_responsables_programa	Tabla Responsables Programa			
Nombre	REQ	PK	FK	Tabla Padre
cod_res_pro	si	si	no	
cod_carg	si	no	si	dbe_cargos

Tabla 4.60: Diccionario de Datos Tabla Responsables Programa

Elaborado por: Amalia Núñez

dbes_responsables_programa	Tabla Responsables Programa			
Nombre	REQ	PK	FK	Tabla Padre
cod_esp	si	no	si	dbes_especialidades
ced_res_pro	si	no	no	
primer_nom_res_pro	si	no	no	
segundo_nom_res_pro	no	no	no	
ape_pat_res_pro	si	no	no	
ape_mat_res_pro	no	no	no	
tel_res_pro	no	no	no	
cel_res_pro	no	no	no	
correo_res_pro	si	no	no	
ciu_res_pro	si	no	no	
prov_res_pro	si	no	no	
dir_res_pro	si	no	no	

Tabla 4.61: Diccionario de Datos Tabla Responsables Programa

Elaborado por: Amalia Núñez

Tabla Títulos Responsable Programa

dbes_titulos_res_programa	Tabla Títulos Responsable Programa			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_tit_resprograma	Código	int		entero largo
cod_res_pro	Código	int		entero largo
nom_tit_resprograma	Nombre	varchar	80	texto

Tabla 4.62: Diccionario de Datos Tabla Títulos Responsable Programa

Elaborado por: Amalia Núñez

dbes_titulos_res_programa	Tabla Títulos Responsable Programa			
Nombre	REQ	PK	FK	Tabla Padre
cod_tit_resprograma	si	si	no	
cod_res_pro	si	no	si	dbes_responsables_programa
nom_tit_resprograma	si	no	no	

Tabla 4.63: Diccionario de Datos Tabla Títulos Responsable Programa

Elaborado por: Amalia Núñez

Tabla Programas Generales

dbe_programas_generales	Tabla Programas Generales			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_pro_gen	Código	int		entero largo
cod_res_pro	Código	int		entero largo
nom_pro	Nombre	varchar	200	texto
fec_pro	Fecha Inicio	datetime		fecha
fec_pro_fin	Fecha Fin	datetime		fecha

Tabla 4.64: Diccionario de Datos Tabla Programas Generales

Elaborado por: Amalia Núñez

dbe_programas_generales	Tabla Programas Generales			
Nombre	REQ	PK	FK	Tabla Padre
cod_pro_gen	si	si	no	
cod_res_pro	si	no	si	dbe_responsables_programa
nom_pro	si	no	no	
fec_pro	si	no	no	
fec_pro_fin	si	no	no	

Tabla 4.65: Diccionario de Datos Tabla Programas Generales

Elaborado por: Amalia Núñez

Tabla Responsables Proyecto

dbe_responsables_proyecto	Tabla Responsables Proyecto			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_resp	Código	int		entero largo
cod_carg	Código	int		entero largo
cod_esp	Código	int		entero largo
ced_resp	Cédula	varchar	10	texto

Tabla 4.66: Diccionario de Datos Tabla Responsables Proyecto

Elaborado por: Amalia Núñez

dbe_responsables_proyecto	Tabla Responsables Proyecto			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
primer_nom_resp	Primer nombre	varchar	25	texto
segundo_nom_resp	Segundo Nombre	varchar	25	texto
ape_pat_resp	Apellido Paterno	varchar	20	texto
ape_mat_resp	Apellido Materno	varchar	20	texto
tel_resp	Teléfono	varchar	12	texto
cel_resp	Celular	varchar	12	texto
correo_resp	Correo	varchar	40	texto
ciu_resp	Ciudad	varchar	30	texto
prov_resp	Provincia	varchar	30	texto
dir_resp	Dirección	varchar	100	texto

Tabla 4.67: Diccionario de Datos Tabla Responsables Proyecto

Elaborado por: Amalia Núñez

dbe_responsables_proyecto	Tabla Responsables Proyecto			
Nombre	REQ	PK	FK	Tabla Padre
cod_resp	si	si	no	
cod_carg	si	no	si	dbe_cargos
cod_esp	si	no	si	dbe_especialidades
ced_resp	si	no	no	
primer_nom_resp	si	no	no	
segundo_nom_resp	no	no	no	
ape_pat_resp	si	no	no	
ape_mat_resp	no	no	no	
tel_resp	no	no	no	
cel_resp	no	no	no	
correo_resp	si	no	no	
ciu_resp	si	no	no	
prov_resp	si	no	no	
dir_resp	si	no	no	

Tabla 4.68: Diccionario de Datos Tabla Responsables Proyecto

Elaborado por: Amalia Núñez

Tabla Títulos Responsable Proyecto

dbe_titulos_res_proyecto	Tabla Títulos Responsable Proyecto			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_tit_proyectos	Código	int		entero largo
cod_resp	Código	int		entero largo
nom_tit_proyectos	Nombre	varchar	80	texto

Tabla 4.69: Diccionario de Datos Tabla Títulos Responsable Proyecto

Elaborado por: Amalia Núñez

dbe_titulos_res_proyecto	Tabla Títulos Responsable Proyecto			
Nombre	REQ	PK	FK	Tabla Padre
cod_tit_proyectos	si	si	no	
cod_resp	si	no	si	dbe_responsables_proyecto
nom_tit_proyectos	si	no	no	

Tabla 4.70: Diccionario de Datos Tabla Títulos Responsable Proyecto

Elaborado por: Amalia Núñez

Tabla Proyectos

dbe_proyectos	Tabla Proyectos			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_proy	Código	int		entero largo
cod_pro_gen	Código	int		entero largo
nom_proy	Nombre	varchar	200	texto
obj_proy	Objetivo	varchar	600	texto
fecha_ini_proy	Fecha Inicio	datetime		fecha
fecha_fin_proy	Fecha Fin	datetime		fecha

Tabla 4.71: Diccionario de Datos Tabla Proyectos

Elaborado por: Amalia Núñez

dbe_proyectos	Tabla Proyectos			
Nombre	REQ	PK	FK	Tabla Padre
cod_proy	si	si	no	
cod_pro_gen	si	no	si	dbe_programas_generales
nom_proy	si	no	no	
obj_proy	no	no	no	
fech_ini_proy	si	no	no	
fech_fin_proy	si	no	no	

Tabla 4.72: Diccionario de Datos Tabla Proyectos

Elaborado por: Amalia Núñez

Tabla Proyectos Responsables

dbe_proyectos_responsables	Tabla Proyectos Responsables			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_pro_res	Código	int		entero largo
cod_resp	Código	int		entero largo
cod_proy	Código	int		entero largo

Tabla 4.73: Diccionario de Datos Tabla Proyectos Responsables

Elaborado por: Amalia Núñez

dbe_proyectos_responsables	Tabla Proyectos Responsables			
Nombre	REQ	PK	FK	Tabla Padre
cod_pro_res	si	si	no	
cod_resp	si	no	si	dbe_responsables_proyecto
cod_proy	si	no	si	dbe_proyectos

Tabla 4.74: Diccionario de Datos Tabla Proyectos Responsables

Elaborado por: Amalia Núñez

Tabla Lugares Actividades

dbe_lugares_actividades	Tabla Lugares Actividades			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_lug_act	Código	int		entero largo
nom_lug_act	Nombre	varchar	90	texto
dir_lug_act	Dirección	varchar	200	texto

Tabla 4.75: Diccionario de Datos Tabla Lugares Actividades

Elaborado por: Amalia Núñez

dbe_lugares_actividades	Tabla Lugares Actividades			
Nombre	REQ	PK	FK	Tabla Padre
cod_lug_act	si	si	no	
nom_lug_act	si	no	no	
dir_lug_act	si	no	no	

Tabla 4.76: Diccionario de Datos Tabla Lugares Actividades

Elaborado por: Amalia Núñez

Tabla Actividades Proyectos

dbe_actividades_proyectos	Tabla Actividades Proyectos			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_act	Código	int		entero largo
cod_proy	Código	int		entero largo
cod_lug_act	Código	int		entero largo
nom_act	Nombre	varchar	600	texto
num_asi_pro_act	Número	int		entero largo
num_asi_act	Número	int		entero largo
meta_act	Meta	decimal	3,2	decimal
informe_act	Informe	varchar	700	texto
evidencia_act	Evidencia	varchar	700	texto

Tabla 4.77: Diccionario de Datos Tabla Actividades Proyectos

Elaborado por: Amalia Núñez

dbe_actividades_proyectos	Tabla Actividades Proyectos			
Nombre	REQ	PK	FK	Tabla Padre
cod_act	si	si	no	
cod_proy	si	no	si	dbe_proyectos
cod_lug_act	si	no	si	dbe_lugares_actividades
nom_act	si	no	no	
num_asi_pro_act	si	no	no	
num_asi_act	si	no	no	
meta_act	si	no	no	
informe_act	no	no	no	
evidencia_act	no	no	no	

Tabla 4.78: Diccionario de Datos Tabla Actividades Proyectos

Elaborado por: Amalia Núñez

Tabla Horarios

dbe_Horarios	Tabla Horarios			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_hor	Código	int		entero largo
cod_act	Código	int		entero largo
fecha	Fecha	datetime		fecha
hor_in	Hora Inicio	datetime		fecha
hor_fin	Hora Fin	datetime		fecha
hombre	Hombre	int		entero largo
mujer	Mujer	int		entero largo
men15	Menor a 15 años	int		entero largo
entre1529	Entre 15 y 29 años	int		entero largo
entre3064	Entre 30 y 64 años	int		entero largo
may64	Mayor a 64 años	int		entero largo
fisica	Discapacidad Física	int		entero largo
psicologica	Discapacidad psicológica	int		entero largo

Tabla 4.79: Diccionario de Datos Tabla Horarios

Elaborado por: Amalia Núñez

dbe_Horarios	Tabla Horarios			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
mental	Discapacidad Mental	int		entero largo
auditiva	Discapacidad Auditiva	int		entero largo
indigenas	Indígena	int		entero largo
mestizos	Mestiza	int		entero largo
blancos	Blanco	int		entero largo
afroamericanos	Afroamericano	int		entero largo
montubios	Montubios	int		entero largo
otros	Otros	int		entero largo

Tabla 4.80: Diccionario de Datos Tabla Horarios

Elaborado por: Amalia Núñez

dbe_Horarios	Tabla Horarios			
Nombre	REQ	PK	FK	Tabla Padre
cod_hor	si	si	no	
cod_act	si	no	si	dbe_actividades_proyectos
fecha	si	no	no	
hor_in	si	no	no	
hor_fin	si	no	no	
hombre	si	no	no	
mujer	si	no	no	
men15	si	no	no	
entre1529	no	no	no	
entre3064	no	no	no	
may64	no	no	no	
fisica	no	no	no	
psicologica	no	no	no	
mental	no	no	no	
auditiva	no	no	no	
indigenas	no	no	no	

Tabla 4.81: Diccionario de Datos Tabla Horarios

Elaborado por: Amalia Núñez

dbe_Horarios	Tabla Horarios			
Nombre	REQ	PK	FK	Tabla Padre
mestizos	no	no	no	
blancos	no	no	no	
afroamericanos	no	no	no	
montubios	no	no	no	
otros	no	no	no	

Tabla 4.82: Diccionario de Datos Tabla Horarios

Elaborado por: Amalia Núñez

Tabla Problemas

dbe_problemas	Tabla Problemas			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_prob	Código	int		entero largo
cod_act	Código	int		entero largo
nom_prob	Nombre	varchar	300	texto

Tabla 4.83: Diccionario de Datos Tabla Problemas

Elaborado por: Amalia Núñez

dbe_problemas	Tabla Problemas			
Nombre	REQ	PK	FK	Tabla Padre
cod_prob	si	si	no	
cod_act	si	no	si	dbe_actividades_proyectos
nom_prob	si	no	no	

Tabla 4.84: Diccionario de Datos Tabla Problemas

Elaborado por: Amalia Núñez

Tabla Soluciones

dbe_soluciones	Tabla Soluciones			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_sol	Código	int		entero largo

Tabla 4.85: Diccionario de Datos Tabla Soluciones

Elaborado por: Amalia Núñez

dbe_soluciones	Tabla Soluciones			
Nombre	Descripción	Tipo de Dato	Tamaño	Formato
cod_prob	Código	int		entero largo
nom_sol	Nombre	varchar	300	texto
des_sol	Descripción	Varchar	100	texto

Tabla 4.86: Diccionario de Datos Tabla Soluciones

Elaborado por: Amalia Núñez

dbe_soluciones	Tabla Soluciones			
Nombre	REQ	PK	FK	Tabla Padre
cod_sol	si	si	no	
cod_prob	si	no	si	dbe_problemas
nom_sol	si	no	no	
des_sol	no	no	no	

Tabla 4.87: Diccionario de Datos Tabla Soluciones

Elaborado por: Amalia Núñez

4.5.4. Diseño de Interfaces

El diseño de la interfaz permite al usuario realizar acciones sobre el Sitio Web que está visitando.

Por lo mismo, se considera parte de la interfaz a sus elementos de identificación, de navegación, de contenidos y de acción.

De acuerdo a los requerimientos del DIBESAU la interfaz se desarrolló en tres partes:

- El Administrador tendrá privilegios altos es decir acceso a todos los recursos del sistema.
- Usuarios con privilegios medios que pueden tener acceso a casi todos recursos del sistema, solo se les limitará el acceso a la administración de usuarios.
- Usuario con bajos privilegios que tendrá acceso únicamente a los reportes.

Así se consigue brindar a los usuarios un entorno sencillo y amigable.

Interfaz Inicio del Sistema



Figura 4.36: Diseño Interfaz Página Inicio
Elaborado por: Amalia Núñez

1. **Url:** Muestra la dirección de la página Web en el navegador.
2. Encabezado del sistema.
3. **Inicio:** Muestra la pantalla de inicio del sistema.
4. Enlaces Estáticos.
5. Calendario.
6. **Iniciar Sesión:** Permite el acceso al sistema.

Interfaz Inicio de Sesión

Esta es la pantalla por la cual accederán tanto los usuarios como el administrador.



Figura 4.37: Diseño Interfaz Inicio de Sesión
Elaborado por: Amalia Núñez

1. Nombre de Usuario.
2. Contraseña.
3. Ingreso al sistema.

Al ingresar como administrador se cuenta con todos los privilegios, el administrador tiene la potestad de crear a los usuarios, darles permisos y darles de baja.

Interfaz Página Principal del Sistema Menú Administrador



Figura 4.38: Diseño Pagina Principal Menú Administrador
Elaborado por: Amalia Núñez

Interfaz Submenús



Figura 4.39: Diseño Interfaz Submenús
Elaborado por: Amalia Núñez

Interfaz Registros



Figura 4.40: Diseño Interfaz Registros
Elaborado por: Amalia Núñez

1. Íconos

EL significado de los íconos es el siguiente:

- Mostrar Datos.
- Nuevo Ingreso.
- Actualizar.

- Eliminar.
- Ver Reporte.

La información corresponde a cada ícono en el orden como se muestran.

2. **Buscar:** La búsqueda es una búsqueda inteligente, se escribe todo o parte de la búsqueda y Enter para buscar.

Interfaz Ingreso Nuevos Registros

Figura 4.41: Diseño Interfaz Ingreso Nuevos Registros
Elaborado por: Amalia Núñez

1. Seleccionar un Registro

Figura 4.42: Diseño Interfaz Seleccionar un Registro
Elaborado por: Amalia Núñez

- Se muestran los datos con el ícono de la lupa o se puede buscar por búsqueda inteligente, se escribe todo o parte de la búsqueda y Enter para buscar.



Figura 4.43: Diseño Interfaz Mostrar Datos
Elaborado por: Amalia Núñez

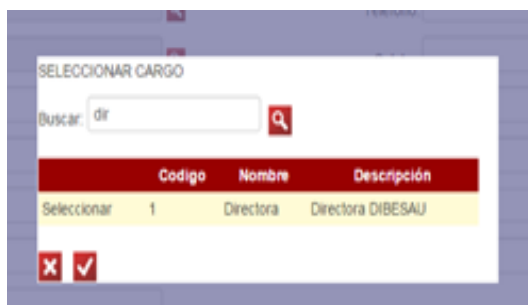


Figura 4.44: Diseño Interfaz Buscar Datos
Elaborado por: Amalia Núñez

- Seleccionar el registro deseado y click en el ícono de visto para terminar con el proceso de selección, caso contrario cancelar con el ícono de la x.



Figura 4.45: Diseño Interfaz Seleccionar un Registro
Elaborado por: Amalia Núñez

2. Cancelar ingreso.
3. Guardar ingreso.
4. Salir sin guardar.

Interfaz Formato para los Campos Fecha

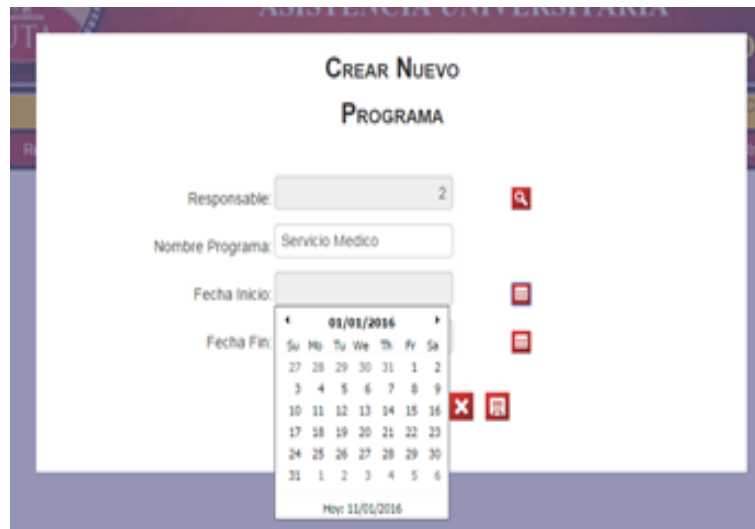


Figura 4.46: Diseño Interfaz Calendario
Elaborado por: Amalia Núñez

Interfaz Modificar Registros



Figura 4.47: Diseño Interfaz Modificar Registros
Elaborado por: Amalia Núñez

Interfaz Eliminar Registros



Figura 4.48: Diseño Interfaz Eliminar Registros
Elaborado por: Amalia Núñez

Interfaz Ver Reporte



Figura 4.49: Diseño Interfaz Ver Reporte
Elaborado por: Amalia Núñez

1. Encabezado de reporte.
2. Cuerpo del Reporte.
3. Exportar reporte en Excel, Pdf, Word.
4. Salir.

Interfaz Grid Anidados



Figura 4.50: Diseño Interfaz de Página Grid Anidados
Elaborado por: Amalia Núñez

4.6. Fase de Implementación

4.6.1. Desarrollo de la Aplicación Web

A continuación se muestra una parte del código, con la que se realizó el proyecto.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.UI;
6 using System.Web.UI.WebControls;
7 using DIBESAUTesis;
8 using System.Data.Objects;
9 using System.Data;
10 using System.IO;
11 using System.Windows.Forms;
12 namespace DIBESAUTesis.Administracion
13 {
14     public partial class ActividadesProyectoFinal2 : System.Web.UI.Page
15     {
16         int aux = 0; //Variable para controlar la búsqueda avanzada
17         int auxabrir = 0;
18         GridView gvbase = new GridView(); //Permite controlar consultas
           a base de datos
19         GridView gvbase1 = new GridView(); //Permite controlar
           consultas a base de datos

```

```

20     private db_dibesauEntities db = new db_dibesauEntities(); //
        Declaramos variable al Entity
21     string codigoactividad;//Variable global para almacenar el
        código seleccionado
22     protected void Page_Load(object sender, EventArgs e)
23     {
24         GridView1.Columns[9].Visible = false;
25         GridView1.Columns[10].Visible = false;
26     }         public void consulta()
27     {
28         ///Mostramos una consulta general de ACTIVIDADES PROYECTO
29         ///mediante un JOIN mostrar información completa
30         var empQuery = from emp in db.dbe_actividades_proyecto
31                         from car in db.dbe_proyectos
32                         from esp in db.dbe_programas_generales
33                         from lug in db.dbe_lugares_actividades
34                         where emp.cod_proy == car.cod_proy
35                         where emp.cod_lug_act == lug.cod_lug_act
36                         where car.cod_pro_gen == esp.cod_pro_gen
37         select new { emp.cod_act, emp.nom_act, emp.num_asi_pro_act,
            emp.num_asi_act, meta = (emp.num_asi_act * 100) / emp.
            num_asi_pro_act, car.nom_proy, esp.nom_pro, lug.
            nom_lug_act, lug.dir_lug_act, emp.informe_act, emp.
            evidencia_act };
38         //GridView1 lugar donde mostramos el Actividad Proyecto
39         GridView1.DataSource = empQuery; //Enlazamos el GridView a
            la consulta
40         GridView1.DataBind(); //Mostrar los Datos
41     }
42     protected void btnBuscar_Click(object sender,
        ImageClickEventArgs e)
43     {
44         if (TextBox1.Text.Trim().Count() == 0)
45         {
46             try
47             {
48                 consulta();
49                 if (GridView1.Rows.Count > 0)
50                 {
51                     lblError.Visible = false;
52                 }
53                 else
54                 {
55                     lblError.Visible = true;
56                 }

```

```

57         }
58     catch (Exception ee)
59     {
60         // MessageBox.Show("error:␣" + ee);
61     }
62 }
63 else
64 {
65     try
66     {
67         //Tomamos el valor del TextBox1
68         string valor = TextBox1.Text;
69         //Realizamos una consulta avanzada
70         var empQuery = from emp in db.
71             dbe_actividades_proyecto
72                 from car in db.dbe_proyectos
73                 from esp in db.
74                     dbe_programas_generales
75                 from lug in db.
76                     dbe_lugares_actividades
77                 where emp.cod_proy == car.cod_proy
78                 where emp.cod_lug_act == lug.
79                     cod_lug_act
80                 where car.cod_pro_gen == esp.
81                     cod_pro_gen
82                 where emp.nom_act.Contains(valor)
83         select new { emp.cod_act, emp.nom_act, emp.
84             num_asi_pro_act, emp.num_asi_act, meta = (
85                 emp.num_asi_act * 100) / emp.num_asi_pro_act
86             , car.nom_proy, esp.nom_pro, lug.nom_lug_act
87             , lug.dir_lug_act, emp.informe_act, emp.
88             evidencia_act };
89
90         GridView1.DataSource = empQuery;
91         GridView1.DataBind();
92         GridView1.DataBind();
93         if (GridView1.Rows.Count > 0)
94         {
95             lblError.Visible = false;
96         }
97         else
98         {
99             lblError.Visible = true;
100        }
101    }

```

```

92         catch (Exception ee)
93         {
94             // MessageBox.Show("error:␣" + ee);
95         }
96     }
97     aux = 0; //devolvemos a cero para saber que sale del
          textBox1
98 }
99
100 protected void btnModificar_Click(object sender,
          ImageClickEventArgs e)
101 {
102     try
103     {
104         //al modificar primero realizamos una consulta para
          recibir los datos.
105         //el código que se envía viene desde la selección del
          GridView1 o General
106         if (GridView1.Rows.Count > 0)
107         {
108             consultaCodigos (Convert.ToInt32(GridView1.Rows[GridView1
          .SelectedRow.RowIndex].Cells[1].Text));
109             if (gvbase.Rows.Count > 0)
110             {
111                 //cargamos los datos obtenidos en las cajas de texto
          correspondientes
112                 txtCodigoModificar.Text = gvbase.Rows[0].Cells[0].
          Text;
113                 txtNomCargoModificar.Text = gvbase.Rows[0].Cells[1].
          Text;
114                 txtNombreEspecialidadModificar.Text = gvbase.Rows
          [0].Cells[2].Text;
115                 txtCodigoCargoModificar.Text = gvbase.Rows[0].Cells
          [1].Text;
116                 txtCodigoEspecialidadModificar.Text = gvbase.Rows
          [0].Cells[2].Text;
117                 txtCedulaModificar.Text = gvbase.Rows[0].Cells[3].
          Text;
118                 txtPNombreModificar.Text = gvbase.Rows[0].Cells[4].
          Text;
119                 //llamamos al Popup Modificar
120                 Button2_ModalPopupExtender.Show();
121             }
122         }
123     }

```

```

124         catch (Exception ee)
125         { // MessageBox.Show("error:␣" + ee);
126         }
127     }
128     protected void btnEliminar_Click(object sender,
        ImageClickEventArgs e)
129     {
130         //al eliminar primero realizamos una consulta para recibir
        los datos.
131         //el código que se envía viene desde la selección del
        GridView1 o General
132         try
133         {
134             if (GridView1.Rows.Count > 0)
135             {
136                 consultaCodigos(Convert.ToInt32(GridView1.Rows[
                    GridView1.SelectedRow.RowIndex].Cells[1].Text));
137                 if (gvbase.Rows.Count > 0)
138                 {
139                     //cargamos los datos obtenidos en las cajas de
                    texto correspondientes
140                     txtCodigoEliminar.Text = gvbase.Rows[0].Cells[0].
                        Text;
141                     txtNombreCargoEliminar.Text = gvbase.Rows[0].
                        Cells[1].Text;
142                     txtNombreEspecialidadEliminar.Text = gvbase.Rows
                        [0].Cells[2].Text;
143                     txtCedulaEliminar.Text = gvbase.Rows[0].Cells[3].
                        Text;
144                     txtNombresEliminar.Text = gvbase.Rows[0].Cells
                        [4].Text;
145                     //Llamamos al Popup de Eliminación
146                     Button3_ModalPopupExtender.Show();
147                 }
148             }
149         }
150         catch (Exception ee)
151         { // MessageBox.Show("error:␣" + ee);
152         }
153     }
154     protected void btnImprimir_Click(object sender,
        ImageClickEventArgs e)
155     {
156         try
157         {

```

```

158         Button4_ModalPopupExtender.Show();
159     }
160     catch (Exception ee)
161     { // MessageBox.Show("error:␣" + ee);
162     }
163 }

```

4.7. Fase de Pruebas

4.7.1. Pruebas de Funcionamiento

Las pruebas es una de las etapas más imprescindibles en la metodología de desarrollo de software, en esta se detecta y corrige los errores en el software.

Al concluir la etapa de desarrollo se realizaron las pruebas, con el fin de depurar errores y mejorar la eficiencia del software. En esta etapa se utilizan dos tipos de pruebas: las de caja blanca y las de caja negra.

4.7.1.1. Pruebas de Caja Blanca

En el desarrollo de un software se denomina caja blanca a las pruebas que se realizan a las funciones internas de la aplicación, las pruebas que se realizaron:

- Revisión de actividades de rutina como el ingreso, edición, eliminación y consulta de datos.
- Se verificó el funcionamiento de la interfaz, el menú sea simple y tenga una visualización agradable para el usuario.
- Evaluar las decisiones del sistema, se evaluó el manejo de errores y las validaciones tanto para el ingreso, modificación y eliminación de datos.

Como las pruebas de caja blanca están ligadas al código fuente estos examinan las funciones, asegurando que durante la prueba se hayan ejecutado todas las sentencias del programa al menos una vez.

Mediante el método de prueba del camino básico se puede obtener la complejidad del diseño y servir como guía para que una función tome una serie de caminos básicos los cuales deben ejecutarse al menos una vez.

Inicio de Sesión

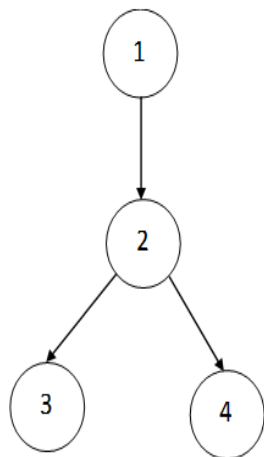


Figura 4.51: Pruebas de Sesión Administrativo
Elaborado por: Amalia Núñez

Proceso	Posibles Caminos
1. Ingreso. 2. Validación. 3. Caso Correcto. 4. Caso Incorrecto.	Camino 1: 1-2-3 Camino 2: 1-2-4

Tabla 4.88: Cuadro de Pruebas de Sesión Administrativo

Ingreso de Datos Personales de Responsables

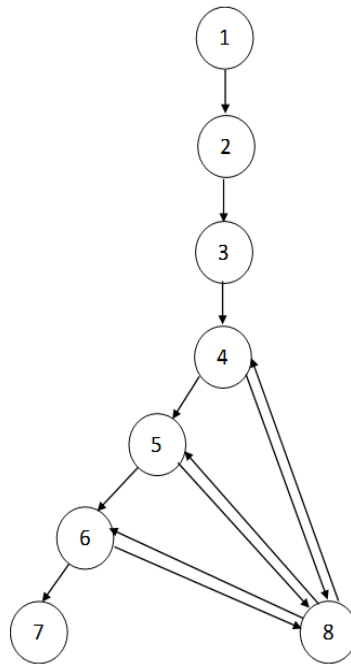


Figura 4.52: Pruebas de Sesión Administrativo
Elaborado por: Amalia Núñez

Proceso	Posibles Caminos
1. Ingreso de número de cédula. 2. Ingreso de información requerida. 3. Ingreso correo electrónico. 4. Validación de cédula. 5. Validación de información requerida. 6. Validación correo electrónico. 7. Almacenamiento. 8. Corrección.	Camino 1: 1-2-3-4-5-6-7 Camino 2: 1-2-3-4-8-4-5-6-7 Camino 3: 1-2-3-4-5-8-5-6-7 Camino 4: 4: 1-2-3-4-5-6-8-6-7

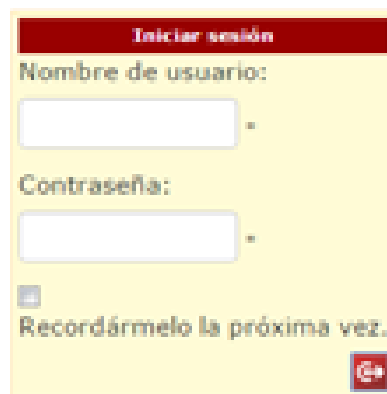
Tabla 4.89: Cuadro de Pruebas de Sesión Administrativo

4.7.1.2. Pruebas de Caja Negra

En el desarrollo de un software se denomina caja negra a las pruebas que tienen como finalidad encontrar cosas que no estén en el contexto del software como:

- Verificar el correcto ingreso de la información.
- Comprobación de roles de los usuarios.
- Entrada y salida de información por parte de los usuarios.
- Controlar que no se visualice información innecesaria para el usuario.

El control de sesiones está dividido en administrativo y modo usuario. Y cuenta con sus respectivas seguridades.



The image shows a login form with a yellow background and a red header bar. The header bar contains the text "Iniciar sesión" in white. Below the header, there are two input fields: "Nombre de usuario:" and "Contraseña:". Each input field has a small red asterisk to its right. Below the password field, there is a checkbox with the text "Recordármelo la próxima vez." to its right. In the bottom right corner of the form, there is a red button with a white right-pointing arrow.

Figura 4.53: Pruebas Iniciar Sección
Elaborado por: Amalia Núñez

Iniciar sesión

Nombre de usuario:
Administrador

Contraseña:

Recordármelo la próxima vez.

El intento de conexión no fue correcto. Inténtelo de nuevo.

Figura 4.54: Pruebas Iniciar Sección
Elaborado por: Amalia Núñez

Control Programa

Revisa que los campos requeridos sean llenados y controla la cédula de identidad.

CREAR NUEVOS DATOS PERSONALES
RESPONSABLES PROGRAMAS

* Cargo:

* Especialidad:

* Cédula:

* Primer Nombre:

Segundo Nombre:

* Apellido Paterno:

* Apellido Materno:

Teléfono:

Celular:

Correo Electrónico:

Ciudad:

Provincia: Azúay

Dirección:

- Cargo es obligatorio
- Especialidad es obligatorio
- La Cédula es obligatoria
- Primer Nombre es obligatorio
- Apellido P. es obligatorio
- Apellido M. es obligatorio

Figura 4.55: Pruebas Ingreso Programas
Elaborado por: Amalia Núñez

Control Actividades

Revisa que los campos requeridos sean llenados.

**CREAR NUEVA
ACTIVIDAD**

* Proyecto:

* Lugar:

* Nombre:

Numero Asistentes Estimado:

- Proyecto obligatorio
- Lugar obligatorio
- Nombre obligatorio

Figura 4.56: Pruebas Ingreso Actividades
Elaborado por: Amalia Núñez

Control Horarios

Revisa que el horario este en el rango de fecha de proyecto, la hora de inicio no puede ser menor a la hora fin y la duración es mínimo de 10 minutos.

localhost:9112 dice:
Fecha no coincide con la fecha del proyecto, entre(02/03/2016-16/03/2016)

Actividad: 1

Fecha: 28/02/2016

Hora Inicio: 0 0

Horario Fin: 0 0

Figura 4.57: Pruebas Ingreso Horarios
Elaborado por: Amalia Núñez

4.7.2. Implantación del Sistema Web

Etapa	Descripción	Involucrados	Fecha
Inducción a usuarios	Presentación y explicación de las fases del proyecto	Directora DIBESAU Técnico DIBESAU Secretaria DIBESAU Amalia Núñez	16/02/2016
Recolección de sugerencias y corrección de errores	Se solicita cambios, correcciones o sugerencias válidas	Directora DIBESAU Técnico DIBESAU Secretaria DIBESAU Amalia Núñez	18/02/2016
Presentación del proyecto con correcciones y sugerencias	Explicación de los cambios realizados	Directora DIBESAU Técnico DIBESAU Secretaria DIBESAU Amalia Núñez	25/02/2016
Prueba del sistema Web por los usuarios	Puesta en marcha del sistema con datos irreales	Directora DIBESAU Técnico DIBESAU Secretaria DIBESAU Amalia Núñez	1/03/2016
Correcciones de Errores encontrados	Con la información ingresada se detecta errores que deben ser corregidos	Directora DIBESAU Técnico DIBESAU Secretaria DIBESAU Amalia Núñez	4/03/2016

Tabla 4.90: Cuadro de Pruebas de Sesión Administrativo

Etapa	Descripción	Involucrados	Fecha
Pruebas de Funcionamiento	Presentación, explicación e ingreso de información	Directora DIBESAU Técnico DIBESAU Secretaria DIBESAU Amalia Núñez	15/03/2016
Pruebas de Aceptación	Pruebas con datos reales con el manual de usuario	Directora DIBESAU Técnico DIBESAU Secretaria DIBESAU Amalia Núñez	18/03/2016
Entrega del Sistema	Entrega formal del Sistema por parte de la Srta. Amalia Núñez a la Dra. Eulalia Pino Loza Directora DIBESAU	Directora DIBESAU Técnico DIBESAU Secretaria DIBESAU Amalia Núñez	21/03/2016

Tabla 4.91: Cuadro de Pruebas de Sesión Administrativo

CAPÍTULO 5

Conclusiones y Recomendaciones

5.1. CONCLUSIONES

- Anteriormente la información al llevarla manualmente se encontraba expuesta a toda vulnerabilidad y el acceso a los datos era complicado e ineficiente, con el desarrollo del sistema Web ahora es posible gestionar y administrar los procesos de registro de responsables, programas, proyectos, actividades, beneficiarios y horarios, se logró obtener facilidad en la confidencialidad, integridad, disponibilidad, manipulación y acceso a los datos.
- Las herramientas seleccionadas en el desarrollo del proyecto permitieron su adecuado acoplamiento lo que ayudó a disminuir el tiempo de implementación de la aplicación, cumpliendo así con los requerimientos establecidos por parte del cliente.
- El uso de ADO.NET Entity Framework como tecnología de acceso a datos, permitió que la aplicación use las clases que se generan a partir del modelo conceptual, ya que abstrae la estructura de la base de datos y todo el acceso y almacenamiento de datos se realiza en base a este modelo conceptual de datos que refleja los objetos de negocio, permitiendo así reducir la cantidad de código y optimizar el tiempo de desarrollo.
- El trabajo con el diseñador ADO.NET Entity Data Model ayudó a generar automáticamente el modelo de entidades a partir de la base de datos, así también a visualizar y modificar gráficamente dicho modelo cuando se producían cambios en la base de datos, generando las clases programables que se emplean en el código de la aplicación.
- Entity Framework es compatible con Language Integrated Query (LINQ), por lo que se utilizó LINQ to Entities para realizar las consultas y obtener resultados en términos de objetos y por ende disminuir la cantidad y

complejidad del código, a diferencia de las consultas tradicionales de SQL a una base relacional, que solo permite guardar tipos de datos primitivos (enteros, cadenas de texto, etc.) por lo que no se pueden manipular de forma directa los objetos de la aplicación y los datos, LINQ permite convertir el modelo relacional en objetos lo que facilita la manipulación y permite que las consultas se hagan de forma genérica sin importar la fuente de datos.

5.2. RECOMENDACIONES

- Se recomienda al Técnico obtener los respaldos necesarios de la base de datos por posibles eventualidades, así también se verifique la fecha de inicio y fin de los programas, ya que de ello depende la asignación correcta de fechas para proyectos y actividades en el sistema Web. Por otra parte los encargados de los proyectos una vez que una actividad se termine deberán ingresar los indicadores en el horario de la actividad, de esto depende la ejecución o no del proyecto y las actividades correspondientes.
- Para la creación de nuevos proyectos en el DIBESAU y en la UTA se sugiere se utilice como tecnología de acceso a datos Entity Framework, ya que esta ayuda en el manejo de consultas y a reducir la cantidad de código, así también esta facilita el mantenimiento del sistema, gracias a que cuenta con herramientas visuales que permiten crear y editar modelos de entidades.
- Se aconseja que se implemente el modulo de participantes en el sistema, el mismo que debería ser enlazado a los datos del UTAMATICO lo que ayudará a que el calculo de los indicadores de los participantes se reflejen automáticamente y en tiempo real.

Bibliografía

- [1] M. de Telecomunicaciones y de la Sociedad de la Información. (2014, Enero) Tecnologías de la información y comunicaciones para el desarrollo. [Online]. Available: <http://www.industrias.ec/archivos/CIG/file/CARTELERA/MINTEL-TIC%20para%20el%20Desarrollo.pdf>
- [2] INEC. (2013) Tecnologías de la información y comunicaciones (tics). [Online]. Available: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/Resultados_principales_140515.Tic.pdf
- [3] D. M. Porras Veloz. (2014) Diseño, desarrollo e implementación de un sistema de control de asistencia del personal administrativo y de servicio para la dirección de talento humano de la universidad central del ecuador. Universidad Central del Ecuador Repositorio Digital. [Online]. Available: <http://www.dspace.uce.edu.ec/handle/25000/2925>
- [4] D. M. Chalán Oviedo. (2014) Sistema de control y adquisiciones de la universidad central del ecuador. Universidad Central del Ecuador Repositorio Digital. [Online]. Available: <http://www.dspace.uce.edu.ec/handle/25000/2847>
- [5] E. J. Pachacama Cruz. (2014, Julio) Análisis, diseño, construcción, e implementación de un sistema web, para la administración académica y capacitación virtual en el instituto niño jesús de praga (injp). Universidad Politécnica Salesiana del Ecuador Repositorio Digital. [Online]. Available: <http://dspace.ups.edu.ec/handle/123456789/7232>
- [6] A. J. Sevilla Alvarado. (2010) Desarrollo e implementación de una aplicación web para la administración de los proyectos de investigación que realiza el ceni de la universidad técnica de ambato. Universidad Técnica de Ambato Repositorio Digital. [Online]. Available: <http://repo.uta.edu.ec/handle/123456789/397>
- [7] A. C. Juarez. Diferencia entre gestión de proyecto y gestión de programa. eHow en Español. [Online]. Available: http://www.ehowenespanol.com/diferencia-gestion-proyecto-gestion-programa-sobre_327183/

- [8] B. E. Lázaro. (2008) Sistema informático. ecured. [Online]. Available: http://www.ecured.cu/index.php/Sistema_inform%C3%A1tico
- [9] IbéricaMultimedia. Sistema informático. IbéricaMultimedia. [Online]. Available: <http://ibericamultimedia.com/secciones-mantenimiento-ordenadores/guia-mantenimiento-sistemas-informaticos/concepto-sistema-informatico>
- [10] r. d. T. d. l. i. y. l. C. A. Universidad de Murcia. Manual básico de creación de páginas web. Universidad de Murcia. [Online]. Available: <https://www.um.es/atika/documentos/html.pdf>
- [11] J. P. Capdevila. Las tecnologías web. Revista Digital Sociedad de la información. [Online]. Available: http://www.sociedadelainformacion.com/9/las_tecnologias_web.htm
- [12] J. E. Pereira. (2010, Enero) Aplicaciones web. mercadeo. [Online]. Available: <http://www.mercadeo.com/blog/2010/01/aplicaciones-web-y-asps/>
- [13] R. G. González. (2010) Arquitectura aplicaciones web. Universidad de Lleida. [Online]. Available: <http://ocw.udl.cat/enginyeria-i-arquitectura/enginyeria-del-software-iii/Continguts/1%20-%20Introduccion/2-Arquitectura.pdf>
- [14] EcuRed. Aplicación web. EcuRed. [Online]. Available: http://www.ecured.cu/index.php/Aplicaci%C3%B3n_web
- [15] J. S. Asenjo. (2012) Servidores de aplicaciones web. Centro Don Bosco Villamuriel de Cerrato. [Online]. Available: <http://www.jorgesanchez.net/web/iaw/iaw1.pdf>
- [16] I. Hoy. (2012) ¿qué es un navegador web? Informatica Hoy. [Online]. Available: <http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-un-navegador-web.php>
- [17] J. S. Asenjo. (2013) Gestión de bases de datos. Centro Don Bosco Villamuriel de Cerrato. [Online]. Available: <http://www.jorgesanchez.net/bd/gbd2012.pdf>
- [18] S. de Informática Universidad de Alicante. Teoría de bases de datos. Universidad de Alicante. [Online]. Available: <http://si.ua.es/es/documentos/documentacion/office/access/teoria-de-bases-de-datos.pdf>
- [19] M. Marqués. (2011) Bases de datos. Universitat Jaume. [Online]. Available: <http://www.uji.es/bin/publ/edicions/bdatos.pdf>
- [20] L. N. de Calidad del Software del Instituto Nacional de Tecnologías de la comunicación (INTECO) España, *Ingeniería del Software: Metodologías y*

- Ciclos de Vida*, primera ed., Instituto Nacional de Tecnologías de la comunicación (INTECO), Marzo 2009. [Online]. Available: https://www.academia.edu/9795641/INGENIERÍA_DEL_SOFTWARE_METODOLOGÍAS_Y_CICLOS_DE_VIDA_Laboratorio_Nacional_de_Calidad_del_Software
- [21] U. P. de Madrid (UPM) Grupo de Sistemas Inteligentes (GSI). El desarrollo del software. Grupo de Sistemas Inteligentes (GSI). [Online]. Available: http://www.gsi.dit.upm.es/~fsaez/intl/libro_complejidad/15-el-desarrollo-del-software.pdf
- [22] R. digital de la Facultad de Ingeniería Universidad Nacional Autónoma de México. Metodologías y procesos de análisis de software. Universidad Nacional Autónoma de México. [Online]. Available: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/175/A5%20Capítulo%202.pdf?sequence=5>
- [23] A. L. G. Guadalupe Ibargüengoitia G, Hanna Oktaba, *Ciclo de Desarrollo de Software*, ser. 19403-1688-29-XI-06, U. Material realizado bajo el convenio 19403-1688-29-XI-06 Microsoft-Facultad de Ciencias, Ed. Primera, Agosto 2010, vol. Uno.
- [24] P. Software and Systems. Modelado de sistemas con uml. Popkin Software and Systems. [Online]. Available: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>
- [25] D. P. Andrés Vignaga. Arquitecturas y tecnologías para el desarrollo de aplicaciones web. Universidad de la República, Facultad de Ingeniería, Instituto de Computación Montevideo, Uruguay. [Online]. Available: http://moodle2.unid.edu.mx/dts_cursos_md/pos/TI/LP/AM/01/Arquitecturas_y_tecnologias_para_el_desarrollo_de_aplicaciones_web.pdf
- [26] F. M. Juan, *Implantación de Aplicaciones Web. CFGS.*, ra-ma ed., RA-MA, Ed. Primera, 2012, vol. Uno.
- [27] culturacion. Frontend y backend: ¿qué son? culturacion. [Online]. Available: <http://culturacion.com/frontend-y-backend-\T1\textquestiondownque-son/>
- [28] EcuRed. Microsoft sql server. EcuRed. [Online]. Available: http://www.ecured.cu/index.php/Microsoft_SQL_Server
- [29] Microsoft. (2007, Noviembre) Información general sobre asp.net. Microsoft. [Online]. Available: <https://msdn.microsoft.com/es-es/library/vstudio/4w3ex9c2%28v=vs.100%29.aspx>

- [30] ——. (2013) Visual studio. Microsoft. [Online]. Available: <https://msdn.microsoft.com/es-es/library/52f3sw5c%28v=vs.100%29.aspx>
- [31] U. de Alicante. Curso .net con c sharp. Universidad de Alicante. [Online]. Available: <http://si.ua.es/es/documentacion/c-sharp/documentos/masterpages/modulo1.pdf>
- [32] Microsoft. (2013) .net framework 4. Microsoft. [Online]. Available: <https://msdn.microsoft.com/es-es/library/vstudio/w0x726c2%28v=vs.100%29.aspx>
- [33] ——. Common language runtime (clr). Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/8bs2ecf4\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/8bs2ecf4(v=vs.110).aspx)
- [34] U. de Alicante. Programación avanzada en .net con c sharp ajax. Universidad de Alicante. [Online]. Available: <http://si.ua.es/es/documentacion/asp-net-programacion-avanzada/documentos/temas/ajax.pdf>
- [35] ASP.NET. Asp.net ajax control toolkit. ASP.NET. [Online]. Available: <http://www.es-asp.net/tutoriales-asp-net/tutorial-5312-5313/que-es-el-asp-net-ajax-control-toolkit.aspx>
- [36] U. de Alicante. Componente reportviewer. Universidad de Alicante. [Online]. Available: <http://si.ua.es/es/documentacion/informes-net/documentos/apuntes-de-referencia/reportviewer.pdf>
- [37] F. P. Víctor. (2011, Mayo) Creando la capa de datos: Entity framework vs linq to sql. Compilando. [Online]. Available: <http://www.compilando.es/?tag=/Entity+Framework>
- [38] C. Payne, *Aprendiendo .NET en 21 Lecciones avanzadas*, P. Educación, Ed. 1st Edition, 2002. [Online]. Available: https://books.google.com.ec/books?id=iXxI7pB8ntkC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- [39] Microsoft. Ventajas de ado.net. Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/3y0bb1zd\(v=vs.90\).aspx](https://msdn.microsoft.com/es-es/library/3y0bb1zd(v=vs.90).aspx)
- [40] H. G. d. B. Osmel Yanes Enriquez. (2011, Diciembre) Mapeo objeto / relacional (orm). Revista Telem@tica. [Online]. Available: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/23/21>
- [41] G. Iván. (2010, Mayo) Introducción a object-relational mapping (orm). Ontuts. [Online]. Available: <http://web.ontuts.com/tutoriales/introduccion-a-object-relational-mapping-orm/>

- [42] L. M. Márquez. Entendiendo linq (language integrated query) con csharp y mono. humansharp. [Online]. Available: <http://www.humansharp.com/docs/TrabajandoLinq.pdf>
- [43] Microsoft. Linq to ado.net (página de portal). Microsoft. [Online]. Available: <https://msdn.microsoft.com/es-es/library/bb397942.aspx>
- [44] ——. Introducción a las consultas linq (csharp). Microsoft. [Online]. Available: <https://msdn.microsoft.com/es-es/library/bb397906.aspx>
- [45] ——. Linq to dataset. Microsoft. [Online]. Available: <https://msdn.microsoft.com/es-es/library/bb386977.aspx>
- [46] J. R. Julien Dollon, *Visual Studio 2010: Desarrollo de aplicaciones web Csharp 4, Framework Entity 4, ASP .NET 4.0, Silverlight 4 y WCF RIA Services*, E. Ediciones, Ed. 1st Edition, 2012.
- [47] K. Gabriel. (2013, Diciembre) Introducción a entity framework. nebaris. [Online]. Available: <http://www.nebaris.com/post/98/introduccion-a-entity-framework>
- [48] Microsoft. Ado.net entity framework. Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/vstudio/bb399572\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/vstudio/bb399572(v=vs.100).aspx)
- [49] S. S. E. Alfredo. (2011, Octubre) Aplicaciones empresariales framework. Escuela Politécnica Nacional. [Online]. Available: <http://bibdigital.epn.edu.ec/bitstream/15000/4295/1/CD-3909.pdf>
- [50] U. de Alicante. Servicio de informática asp.net mvc 2 framework. Universidad de Alicante. [Online]. Available: <http://si.ua.es/es/documentacion/asp-net-mvc-2/ado-net-entity-framework/ado-net-entity-framework.html>
- [51] Microsoft. Información general de entity framework. Microsoft. [Online]. Available: <https://msdn.microsoft.com/es-es/library/bb399567%28v=vs.110%29.aspx>
- [52] ——. Ado.net data providers. Microsoft. [Online]. Available: <https://msdn.microsoft.com/en-us/data/dd363565.aspx>
- [53] ——. Ado.net entity framework. Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/vstudio/bb399572\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/vstudio/bb399572(v=vs.100).aspx)
- [54] ——. Entity data model. Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/ee382825\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/ee382825(v=vs.110).aspx)
- [55] ——. Modelado de datos en entity framework. Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/bb399183\(v=vs.90\).aspx](https://msdn.microsoft.com/es-es/library/bb399183(v=vs.90).aspx)

- [56] U. Z. Castro, *ADO.NET Entity Framework 4.1 Aplicaciones y servicios centrados en datos*, S. L. Krasis Consulting, Ed. Primera edición en español, 2011.
- [57] Microsoft. Asignar un modelo conceptual a un modelo de almacenamiento. Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/bb399232\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/bb399232(v=vs.100).aspx)
- [58] ——. Usar las herramientas de entity data model. Microsoft. [Online]. Available: [https://msdn.microsoft.com/library/ee518968\(v=vs.100\).aspx](https://msdn.microsoft.com/library/ee518968(v=vs.100).aspx)
- [59] ——. Proveedor de entityclient para entity framework. Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/bb738561\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/bb738561(v=vs.100).aspx)
- [60] ——. Información general de servicios de objeto (entity framework). Microsoft. [Online]. Available: [https://msdn.microsoft.com/es-es/library/bb386871\(v=vs.90\).aspx](https://msdn.microsoft.com/es-es/library/bb386871(v=vs.90).aspx)
- [61] M. I. S. S. Xavier Ferré Grau. Desarrollo orientado a objetos con uml. Facultad de Informática UPM. [Online]. Available: <http://www.uv.mx/personal/maymendez/files/2011/05/umlTotal.pdf>

Anexos y Apéndices

Anexo A

Manual de Usuario

Universidad Técnica de Ambato

Facultad de Ingeniería en Sistemas, Electrónica e Industrial



Sistema para Gestión de Programas de Servicios Comunitarios para el DIBESAU de la UTA

A.1. Introducción

Este manual permitirá aprender a utilizar todas las funcionalidades del Sistema de Gestión de Programas de Servicios Comunitarios para el DIBESAU de la UTA.

El Sistema permite gestionar los programas de servicios comunitarios desde su proceso de registro hasta su proceso ejecución en los cuales consta:



Figura A.1: Menú Administrador
Elaborado por: Amalia Núñez

- Inicio
- Responsables/Datos Laborales
 - Ingreso de Cargos.
 - Ingreso de Especialidades.
- Programas
 - Ingreso de Responsables/Datos Personales.
 - Ingreso de Responsables/Datos Académicos.
 - Ingreso Nuevos Programas.
- Proyectos
 - Ingreso de Responsables/Datos Personales.
 - Ingreso de Responsables/Datos Académicos.
 - Ingreso Nuevos Proyectos.
- Actividades
 - Ingreso de Lugares.
 - Ingreso de Actividad/Proyecto.
 - Calculo de Indicadores.
 - Ingreso de Problemas y Soluciones.
- Informes

A.2. Acceso a la Aplicación

El sistema para gestión de programas de servicios comunitarios, es una aplicación Web que puede ser accedida desde cualquier navegador de internet.

La dirección URL es (Definir la Dirección)

El usuario debe ingresar a un navegador Web (Chrome, Firefox, Opera) y escribir la dirección URL del sitio. Una vez cargada la página se visualiza una pantalla de inicio del sistema.



Figura A.2: Pantalla de Inicio del Sistema
Elaborado por: Amalia Núñez

Dar click en Iniciar Sesión para acceder al sistema y aparecerá la pantalla de ingreso al sistema.



Figura A.3: Pantalla de Ingreso al Sistema
Elaborado por: Amalia Núñez

Para comenzar a utilizar el Sistema en el usuario debe ingresar su nombre de usuario y contraseña y presionar el botón “Ingresar”.

Si el ingreso fue con privilegios de administrador, se muestra el siguiente menú:



Figura A.4: Menú Administrador
Elaborado por: Amalia Núñez

A.2.1. Funcionalidades del Sistema

- Ingreso de cargos de los representantes de programas y proyectos.
- Ingreso de especialidades de los representantes de programas y proyectos.

- Ingreso de responsables programas responsables/datos personales.
- Ingreso de responsables programas responsables/datos académicos.
- Ingreso de nuevo programa y selección del representante.
- Ingreso de responsables proyecto responsables/datos personales.
- Ingreso de responsables proyectos responsables/datos académicos.
- Ingreso de nuevo proyecto y asignación de los representantes.
- Ingreso de los lugares en los que se realizarán las actividades.
- Ingreso de actividades/proyecto.
- Asignación de horarios a la actividad.
- Asignación de beneficiarios en la actividad y horario.
- Cálculo de indicadores.
- Asignación de problemas encontrados en las actividades.
- Asignación de soluciones a los problemas encontrados en las actividades.
- Informes.

A.3. Guía de Usuario

Pantalla Menú Responsables/Datos Laborales

En esta pantalla se encuentra la información de cargos y especialidades, la misma que permite, consultar, ingresar nuevo, modificar, y acceder al informe, esta información posteriormente se seleccionará para los responsables de programas y proyectos.



Figura A.5: Pantalla Menú Responsables/Datos Laborales
Elaborado por: Amalia Núñez

Pantalla Responsables/Datos Personales Cargos



Figura A.6: Pantalla de Cargos
Elaborado por: Amalia Núñez

1. Íconos

EL significado de los íconos es el siguiente:

- Mostrar Datos.
- Nuevo.
- Actualizar.

- Eliminar.
- Imprimir informe.

La información corresponde a cada ícono en el orden como se muestran.

2. **Buscar:** La búsqueda se realiza por el nombre del cargo, se escribe todo o parte del nombre y Enter para buscar.

Pantalla de Ingreso de un Nuevo Registro de Cargo

Figura A.7: Pantalla de Ingreso de un Nuevo Registro de Cargo
Elaborado por: Amalia Núñez

Proceso de Modificar un Registro de Cargo

Seleccionar el registro deseado y click en el ícono de modificar.

	Código	Nombre	Descripción
Seleccionar	1	Directora	Directora DIBESAU
Seleccionar	2	Jefe	Jefe de Medicos

Figura A.8: Proceso de Modificar un Registro de Cargos
Elaborado por: Amalia Núñez

Pantalla de Modificar un Registro de Cargo



Figura A.9: Pantalla de Modificar un Registro de Cargos
Elaborado por: Amalia Núñez

Pantalla de Eliminación del Registro de Cargo



Figura A.10: Pantalla de Eliminación del Registro de Cargos
Elaborado por: Amalia Núñez

NOTA: El mismo proceso se lo realiza con Especialidades.



Figura A.11: Pantalla Registro de Especialidades
Elaborado por: Amalia Núñez

Pantalla Menú de Programas



Figura A.12: Pantalla Menú de Programas
Elaborado por: Amalia Núñez

Pantalla Programas Responsables/Datos Personales

En esta pantalla se encuentra la información personal de los responsables de los programas, la cual permite, consultar, ingresar nuevo, modificar, y acceder al informe, esta información posteriormente se seleccionará para asignar un responsable al Programa.



Figura A.13: Opción Responsables/Datos Personales
Elaborado por: Amalia Núñez



Figura A.14: Pantalla Responsables/Datos Personales
Elaborado por: Amalia Núñez

Pantalla Programas Nuevo Registro Responsables/Datos Personales

Figura A.15: Pantalla de Nuevo Registro Responsables/Datos Personales
Elaborado por: Amalia Núñez

1. Seleccionar el cargo que tiene el responsable.

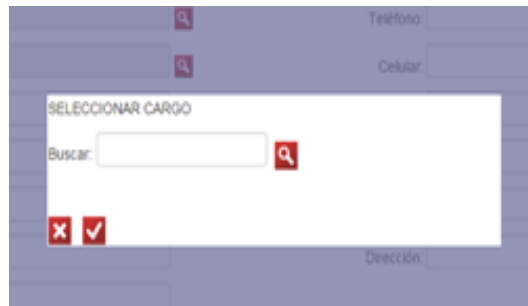


Figura A.16: Pantalla Seleccionar el Cargo
Elaborado por: Amalia Núñez

- Se muestran los datos con el ícono de la lupa o se puede buscar por nombre del cargo, poner todo o parte del nombre y Enter.



Figura A.17: Mostrar Datos de Cargos
Elaborado por: Amalia Núñez

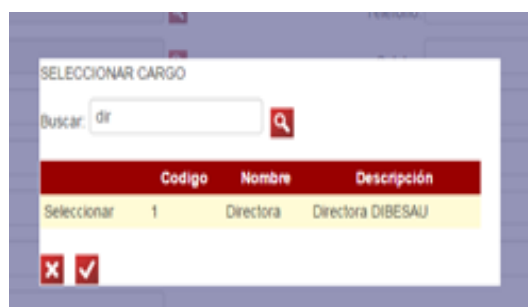


Figura A.18: Buscar Datos de Cargos
Elaborado por: Amalia Núñez

- Seleccionar el registro deseado y click en el ícono de visto para terminar con el proceso de selección, caso contrario cancelar con el ícono de la x.



Figura A.19: Seleccionar el Cargo
Elaborado por: Amalia Núñez

2. De la misma manera se realiza la selección de la especialidad.
3. Cancelar Ingreso.
4. Guardar Ingreso.
5. Salir sin guardar.

El registro de datos personales de los responsables de los programas cuenta con varias restricciones como lo es:

- La cédula no puede ser incorrecta y es obligatoria.

Figura A.20: Error Cédula Incorrecta
Elaborado por: Amalia Núñez

- Campos obligatorios.

**CREAR NUEVOS DATOS PERSONALES
RESPONSABLES PROGRAMAS**

Cargo:

Especialidad:

Cédula:

* Primer Nombre:

Segundo Nombre:

* Apellido Paterno:

* Apellido Materno:

Teléfono:

Celular:

Correo Electrónico:

Ciudad:

Provincia:

Dirección:

- Primer Nombre es obligatorio
- Apellido P. es obligatorio
- Apellido M. es obligatorio

Figura A.21: Error Campos Requeridos
Elaborado por: Amalia Núñez

- El correo electrónico tiene que ser valido.

**CREAR NUEVOS DATOS PERSONALES
RESPONSABLES PROGRAMAS**

Cargo:

Especialidad:

Cédula:

Primer Nombre:

Segundo Nombre:

Apellido Paterno:

Apellido Materno:

Teléfono:

Celular:

* Correo Electrónico:

Ciudad:

Provincia:

Dirección:

- Correo Electrónico Invalido

Figura A.22: Error Correo Electrónico
Elaborado por: Amalia Núñez

Si toda la información está bien se procederá a guardar.

RESPONSABLES PROGRAMAS

BUSCAR

Apellido: Cédula:

Código	Cargo	Especialidad	Cédula	Nombre	Segundo	Apellido_Paterno	Apellido_Materno	Teléfono	Celular
Seleccionar 1	Jefe	Ginecología	1804800748	Carolina	Estefanía	Muñoz	Camacho	0122745	18048007

Figura A.23: Datos Guardados Responsables Programas Datos Personales
Elaborado por: Amalia Núñez

Pantalla Programas Modificar Responsables/Datos Personales

MODIFICAR

DATOS PERSONALES RESPONSABLE DE PROGRAMAS

Código: Apellido Materno:

Cargo: Teléfono:

Especialidad: Celular:

Cédula: Correo Electrónico:

Primer Nombre: Ciudad:

Segundo Nombre: Provincia:

Apellido Paterno: Dirección:

Figura A.24: Pantalla de Modificación Responsables Programas Datos Personales
Elaborado por: Amalia Núñez

Pantalla Programas Eliminar Responsables/Datos Personales

ELIMINAR RESPONSABLE PROGRAMA

Código:

Cargo:

Especialidad:

Cédula:

Nombre:

Apellido:

Figura A.25: Pantalla de Eliminación Responsables Programas Datos Personales
Elaborado por: Amalia Núñez

Ver Reporte y Exportar a Pdf



Figura A.26: Ver Reporte y Exportar a Pdf
Elaborado por: Amalia Núñez

Ver Reporte en Pdf

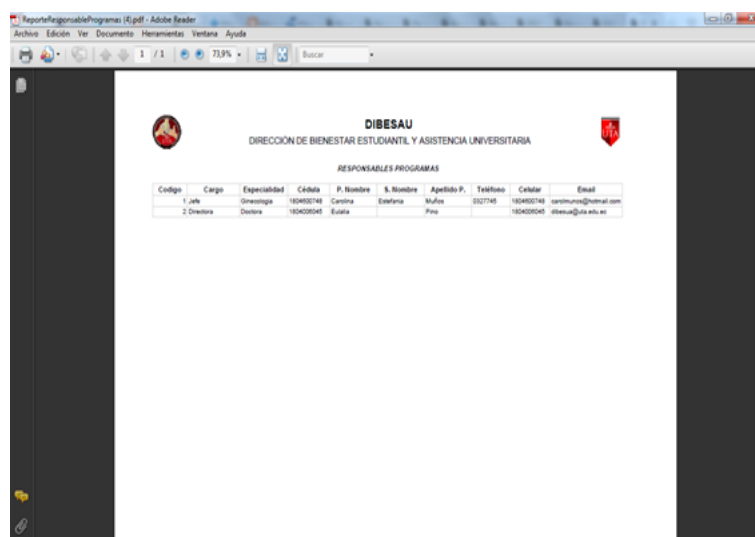


Figura A.27: Ver Reporte en Pdf
Elaborado por: Amalia Núñez

Pantalla Programas Responsables/Datos Académicos



Figura A.28: Opción Responsables/Datos Académicos
Elaborado por: Amalia Núñez



Figura A.29: Pantalla Responsables/Datos Académicos
Elaborado por: Amalia Núñez

Pantalla de Ingreso Nuevo Dato Académico del Responsable del Programa



Figura A.30: Pantalla de Ingreso Nuevo Dato Académico
Elaborado por: Amalia Núñez

- Seleccionar el responsable del programa al cual corresponde el título.

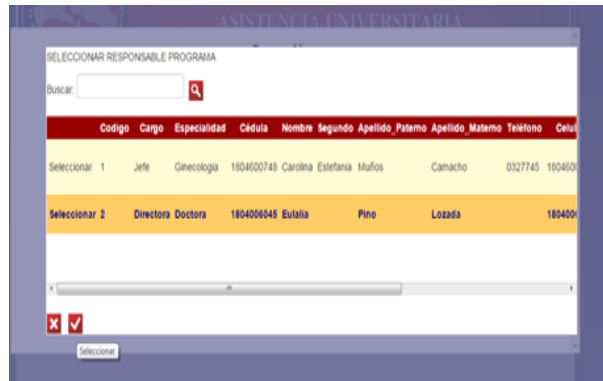


Figura A.31: Seleccionar el Responsable del Programa al que Corresponde el Titulo
Elaborado por: Amalia Núñez

- Ingresar la información y guardar.



Figura A.32: Ingresar la información y Guardar
Elaborado por: Amalia Núñez

- Información guardada correctamente.



Figura A.33: Información Guardada Correctamente
Elaborado por: Amalia Núñez

Pantalla de Modificar Responsables/Datos Académicos

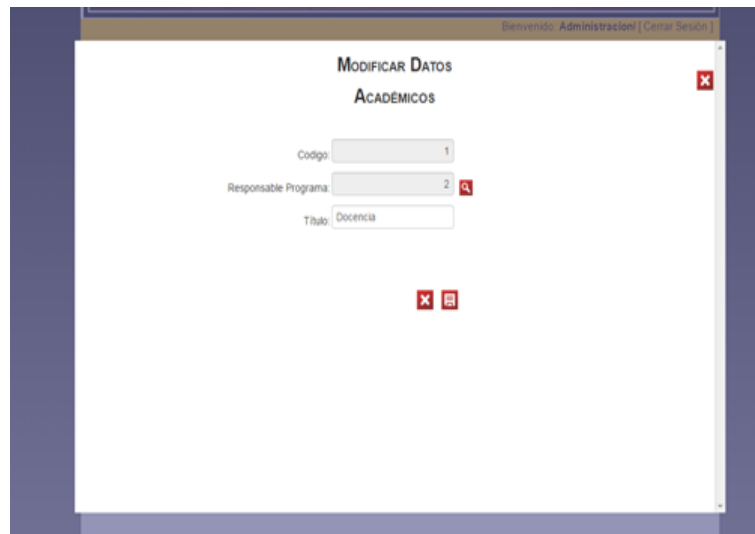


Figura A.34: Pantalla de Modificar Responsables/Datos Académicos
Elaborado por: Amalia Núñez

Pantalla de Eliminar de Responsables/Datos Académicos



Figura A.35: Pantalla de Eliminar Responsables/Datos Académicos
Elaborado por: Amalia Núñez

Ver Reporte y Exportar a Pdf



Figura A.36: Ver Reporte y Exportar a Pdf
Elaborado por: Amalia Núñez

Pantalla Nuevos Programas



Figura A.37: Opción Nuevos Programas
Elaborado por: Amalia Núñez



Figura A.38: Pantalla de Nuevos Programas
Elaborado por: Amalia Núñez

Pantalla Ingreso Nuevos Programas



Figura A.39: Pantalla Ingreso Nuevos Programas
Elaborado por: Amalia Núñez

1. Seleccionar Responsable del Programa.

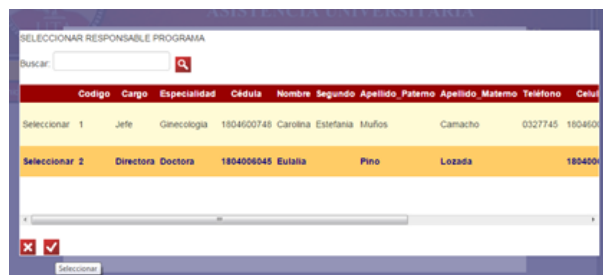


Figura A.40: Seleccionar Responsable del Programa
Elaborado por: Amalia Núñez

2. Seleccionar la Fecha de Inicio del Programa.

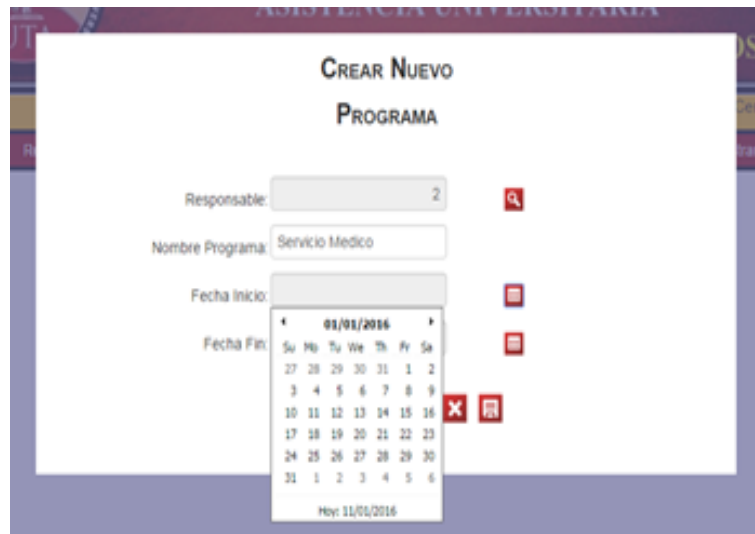


Figura A.41: Seleccionamos la Fecha de Inicio del Programa
Elaborado por: Amalia Núñez

3. Seleccionar la Fecha de Fin del Programa.

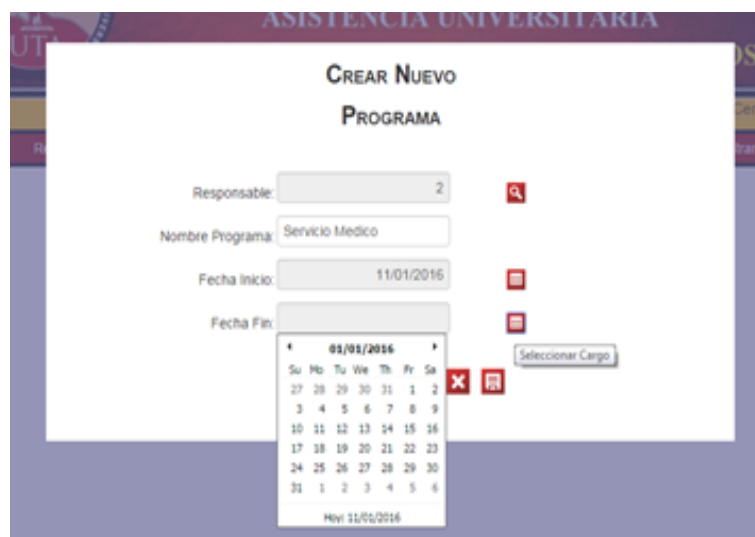


Figura A.42: Seleccionamos la Fecha de Fin del Programa
Elaborado por: Amalia Núñez

Las fechas tienen las siguientes restricciones:

- La fecha de inicio no puede ser menor que la fecha fin.
- La fecha fin no puede ser la misma de la de inicio.



Figura A.43: Error La Fecha de Inicio no Puede ser Menor o Igual a la fecha Fin
Elaborado por: Amalia Núñez

Nota: Un programa no puede tener la fecha de duración menos a la de un día.

4. Guardar.
5. Cancelar.

Si la información es correcta se guardan los datos.



Figura A.44: Mostrar Datos Guardados
Elaborado por: Amalia Núñez

Pantalla de Modificar Programas

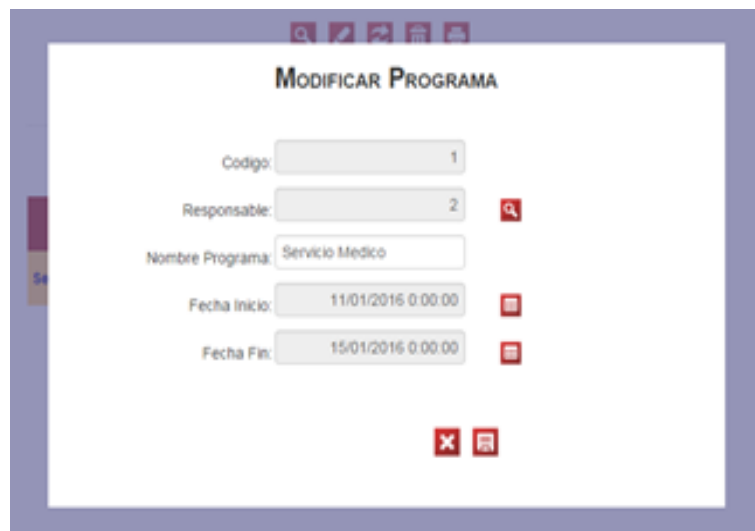


Figura A.45: Pantalla Modificar Programas
Elaborado por: Amalia Núñez

Pantalla de Eliminar Programas



Figura A.46: Pantalla Eliminar Programas
Elaborado por: Amalia Núñez

Ver Informe y Exportar a Pdf

IMPRIMIR

DIBESAU
DIRECCIÓN DE BIENESTAR ESTUDIANTIL Y ASISTENCIA
UNIVERSITARIA

PROGRAMAS

Codigo	Nombre	Fecha Inicio	Fecha Fin	Cédula	Nombre	Apellido
1	Servicio Medico	11/01/2016 0:00:00	15/01/2016 0:00:00	1804000045	Eulalia	Pino

Figura A.47: Ver Informe
Elaborado por: Amalia Núñez

Pantalla Menú de Proyectos

DIBESAU
DIRECCIÓN DE BIENESTAR ESTUDIANTIL Y
ASISTENCIA UNIVERSITARIA
PROGRAMAS DE SERVICIOS COMUNITARIOS

Bienvenido **Administración!** [Cerrar Sesión]

Inicio Responsables/Datos Laborales Programas **Proyectos** Actividades Informes Administrar Usuarios

- Responsables/Datos Personales
- Responsables/Datos Académicos
- Nuevos Proyectos

DIBESAU
 BECAS
 UTA
 CALENDARIO

enero de 2016

lun mar mié jue vie sáb dom
 28 29 30 31 1 2 3
 4 5 6 7 8 9 10
 11 12 13 14 15 16 17
 18 19 20 21 22 23 24
 25 26 27 28 29 30 31

Figura A.48: Pantalla Menú de Proyectos
Elaborado por: Amalia Núñez

Pantalla Proyectos Responsables/Datos Personales



Figura A.49: Pantalla Proyectos Responsables/Datos Personales
Elaborado por: Amalia Núñez

Pantalla Proyectos Responsables/Datos Académicos



Figura A.50: Pantalla Proyectos Responsables/Datos Académicos
Elaborado por: Amalia Núñez

Nota: El mismo proceso que se realizó en Programas para Responsables/Datos Personales y Responsables/Datos Académicos, se lo realiza en Proyectos ya que se ingresan los mismos datos, con la diferencia en que estos corresponden a los responsables que se asignarán al proyecto.

Pantalla Nuevos Proyectos



Figura A.51: Opción Nuevos Proyectos
Elaborado por: Amalia Núñez



Figura A.52: Pantalla Nuevos Proyectos
Elaborado por: Amalia Núñez

Pantalla Ingreso Nuevos Proyectos



Figura A.53: Pantalla Ingreso Nuevos Proyectos
Elaborado por: Amalia Núñez

1. Seleccionar el Programa al que Corresponde el Proyecto.



Figura A.54: Seleccionar el Programa al que Corresponde el Proyecto.
Elaborado por: Amalia Núñez

2. Seleccionar la Fecha de Inicio del Proyecto.
3. Seleccionar la Fecha de Fin del Proyecto.

Las fechas tienen que estar en el rango de las fechas del programa seleccionado.



Figura A.55: Error Fechas
Elaborado por: Amalia Núñez

4. Guardar.
5. Cancelar.

Nota: Todos los Campos son Obligatorios.

CREAR NUEVO PROYECTO

* Programa:

* Nombre Proyecto:

* Objetivo:

* Fecha Inicio:

* Fecha Fin:

- Responsable Programa obligatorio
- Nombre Programa obligatorio
- Objetivo obligatorio
- Fecha Inicio obligatorio
- Fecha Fin obligatorio

Figura A.56: Error Campos Obligatorios
Elaborado por: Amalia Núñez

Ver Informe y Exportar a Pdf

IMPRESOR

DIBESAU
DIRECCIÓN DE BIENESTAR ESTUDIANTIL Y ASISTENCIA
UNIVERSITARIA

PROYECTOS

Código	Proyecto	Objetivo	Fecha I.	Fecha F.	Programa
1	Charla sobre el VIH	Que los estudiantes conozcan los riesgos del VIH	11/01/2016	15/01/2016	Servicio Medico

Figura A.57: Ver Informe
Elaborado por: Amalia Núñez

Una vez ingresada la información de proyectos se procede asignar los responsables a proyectos.

Nota: Para poder asignar un nuevo responsable a proyecto primero hay que seleccionar el proyecto al que se asignará dicho responsable.

DIBESAU
DIRECCIÓN DE BIENESTAR ESTUDIANTIL Y ASISTENCIA UNIVERSITARIA
PROGRAMAS DE SERVICIOS COMUNITARIOS

Bienvenido: Administración! [Cerrar Sesión]

Inicio Responsables/Delos Laborales Programas Proyectos Actividades Participantes Administrar Usuarios

PROYECTOS

Buscar:

Codigo	Nombre	Objetivo	F. Inicio	F. Fin	Programa
Seleccionar 1	Charla sobre el VIH	Que los estudiantes conozcan los riesgos del VIH	11/01/2016 0:00:00	15/01/2016 0:00:00	Servicio Medico

RESPONSABLES PROYECTOS

No existen resultados [Ingresar Nuevo](#)

Figura A.58: Pantalla Responsable Proyecto
Elaborado por: Amalia Núñez

Pantalla Asignar Responsable a Proyecto

CREAR NUEVO RESPONSABLE/PROYECTO

Codigo Proyecto:

Codigo Responsable:

Cargo:

Especialidad:

Cédula:

Nombre:

Apellido:

Figura A.59: Pantalla Asignar Responsable Proyecto
Elaborado por: Amalia Núñez

- Seleccionar el responsable del proyecto.



Figura A.60: Seleccionar el Responsable del Proyecto
Elaborado por: Amalia Núñez

- Al seleccionar el responsable del proyecto se cargara automáticamente la información.

CREAR NUEVO RESPONSABLE/PROYECTO

Codigo Proyecto:

Codigo Responsable:

Cargo:

Especialidad:

Cédula:

Nombre:

Apellido:

Buttons: [X] [📄]

Figura A.61: Ingresar la información y Guardar
Elaborado por: Amalia Núñez

- Información guardada correctamente.

PROYECTOS

Buttons: [🔍] [✓] [📄] [🔄]

Buscar:

Codigo	Nombre	Objetivo	F. Inicio	F. Fin	Programa
Seleccionar 1	Charla sobre el VIH	Que los estudiantes conozcan los riesgos del VIH	11/01/2016	15/01/2016	Servicio Medico

RESPONSABLES PROYECTOS

Buttons: [🔍] [✓] [📄] [🔄]

Codigo	Cargo	Especialidad	Cédula	Nombre	Apellido	Teléfono
Seleccionar 1	Jefe	Ginecología	1804600748	Carmen	Caceres	

Figura A.62: Información Guardada Correctamente
Elaborado por: Amalia Núñez

Pantalla Eliminar Responsable de Proyecto



Figura A.63: Pantalla Eliminar Responsable/Proyecto
Elaborado por: Amalia Núñez

Pantalla Menú Actividades



Figura A.64: Pantalla Menú Actividades
Elaborado por: Amalia Núñez

Pantalla Actividades Lugares



Figura A.65: Opción Actividades Lugares
Elaborado por: Amalia Núñez



Figura A.66: Pantalla Lugares Actividades
Elaborado por: Amalia Núñez

Pantalla de Ingreso Nuevo Registro de Lugares



Figura A.67: Pantalla de Ingreso Nuevo Registro de Lugares
Elaborado por: Amalia Núñez

Pantalla de Modificar un Registro de Lugares



Figura A.68: Pantalla de Modificar un Registro de Lugares
Elaborado por: Amalia Núñez

Pantalla de Eliminación del Registro de Lugares



Figura A.69: Pantalla de Eliminación del Registro de lugares
Elaborado por: Amalia Núñez

Ver Informe y Exportar a Pdf



Figura A.70: Ver Informe
Elaborado por: Amalia Núñez

Pantalla Actividades Proyecto



Figura A.71: Opción Actividades Lugares
Elaborado por: Amalia Núñez



Figura A.72: Pantalla Actividades Proyectos
Elaborado por: Amalia Núñez

Pantalla de Ingreso Nuevo Actividades Proyectos

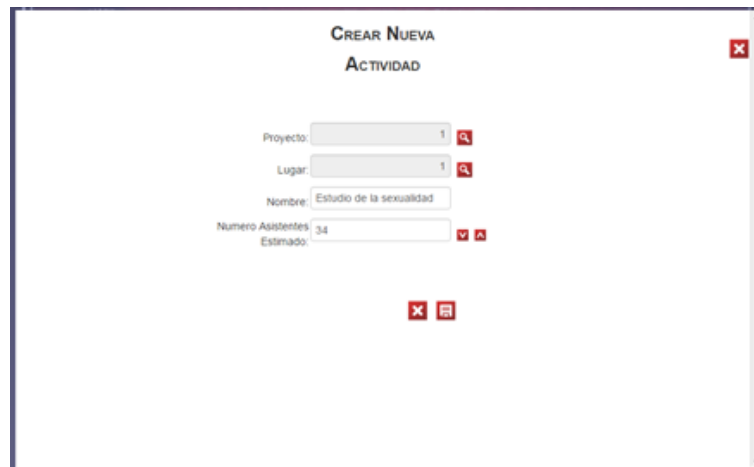


Figura A.73: Pantalla de Ingreso Nuevo Registro de Actividades Proyecto
Elaborado por: Amalia Núñez

- Seleccionar el proyecto al que pertenece la actividad.



Figura A.74: Seleccionar el Proyecto
Elaborado por: Amalia Núñez

- Seleccionar el lugar donde se realizará la actividad.



Figura A.75: Seleccionar el Lugar
Elaborado por: Amalia Núñez

- Información guardada correctamente.

The screenshot shows the 'ACTIVIDADES PROYECTOS' page with a search bar and a table of activities. Below the table is a 'HORARIOS' section with a search bar and the text 'No existen resultados'.

Código	Actividad	Asi. Est.	Asi. Act.	Meta	Proyecto	Programa	Lugar	Dirección	Informe	Evidencia
Seleccionar 1	Charla	50	1	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 2	charla	100	0	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 3	Charla	20	0	0,00	Socializacion	Becas	Auditorio FISEI	Huachi Chico		
Seleccionar 4	Estudio de la sexualidad	34	0	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		

Figura A.76: Información Guardada Correctamente
Elaborado por: Amalia Núñez

Una vez ingresada la información de actividades proyectos se procede asignar el horario a la actividad.

Nota: Para poder asignar un nuevo horario a la actividad primero hay que seleccionar la actividad al que se asignará dicho horario.

The screenshot shows the 'ACTIVIDADES PROYECTOS' page with a search bar and a table of activities. Below the table is a 'HORARIOS' section with a search bar, an 'Ingresar Nuevo' button, and the text 'No existen resultados'.

Código	Actividad	Asi. Est.	Asi. Act.	Meta	Proyecto	Programa	Lugar	Dirección	Informe	Evidencia
Seleccionar 1	Charla	50	1	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 2	charla	100	0	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 3	Charla	20	0	0,00	Socializacion	Becas	Auditorio FISEI	Huachi Chico		
Seleccionar 4	Estudio de la sexualidad	34	0	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		

Figura A.77: Pantalla Horarios
Elaborado por: Amalia Núñez

Pantalla de Ingreso Nuevo Horario Actividad



Figura A.78: Pantalla de Ingreso Nuevo Horario Actividad
Elaborado por: Amalia Núñez

- Seleccionar la fecha de la actividad.



Figura A.79: Seleccionar el Proyecto
Elaborado por: Amalia Núñez

La fecha debe estar en el el rango de fecha del proyecto.

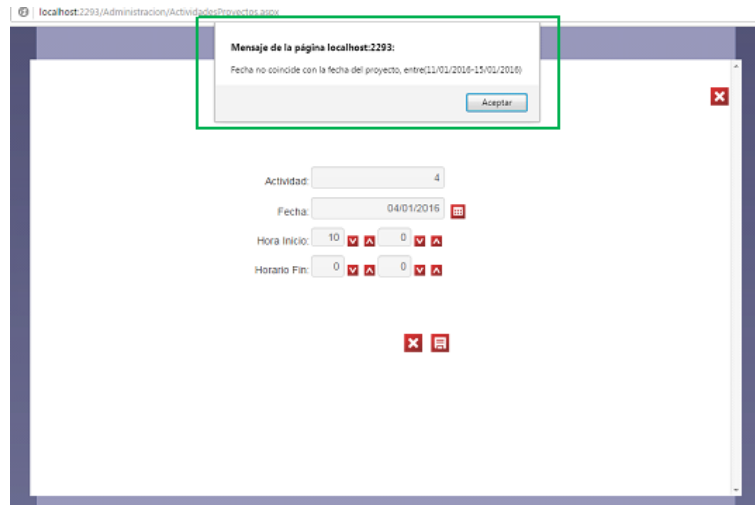


Figura A.80: Seleccionar el Lugar
Elaborado por: Amalia Núñez

- Información guardada correctamente.

Código	Actividad	ANI	ANI	ANI	Meta	Proyecto	Programa	Lugar	Dirección	Informe	Evidencia
Seleccionar 1	Charla	10	1	0,00	0,00	Charla sobre el VIH	Servicio Médico FISD	Auditorio Huachi Chico			
Seleccionar 2	charla	100	0	0,00	0,00	Charla sobre el VIH	Servicio Médico FISD	Auditorio Huachi Chico			
Seleccionar 3	Charla	20	0	0,00	0,00	Socialización	Becas FISD	Auditorio Huachi Chico			
Seleccionar 4	Estudio de la sexualidad	34	0	0,00	0,00	Charla sobre el VIH	Servicio Médico FISD	Auditorio Huachi Chico			

HORARIOS			
Código	Fecha	Hora Inicio	Hora Fin
Seleccionar 4	12/01/2016 0:00:00	10:00:00	10:10:00

Figura A.81: Información Guardada Correctamente
Elaborado por: Amalia Núñez

Pantalla Modificar Horario Actividad

The screenshot shows a web form titled "MODIFICAR HORARIO" with a close button (X) in the top right corner. The form contains the following fields: "Codigo" with the value "6", "Actividad" with the value "4", "Fecha" with the value "12/01/2016" and a calendar icon, "Hora Inicio" with the value "10" and a time selection interface, and "Horario Fin" with the value "10" and a time selection interface. At the bottom center, there are two buttons: a red "X" button and a red icon button.

Figura A.82: Pantalla Modificar Horario Actividad
Elaborado por: Amalia Núñez

Pantalla Eliminar Horario Actividad

The screenshot shows a web form titled "ELIMINAR HORARIO" with a close button (X) in the top right corner. The form contains the following fields: "Codigo" with the value "6", "Actividad" with the value "4", "Fecha" with the value "12/01/2016 0:00:00", "Hora Inicio" with the value "10:00:00", and "Horario Fin" with the value "10:10:00". At the bottom center, there are two buttons: a red "X" button and a red icon button.

Figura A.83: Pantalla Eliminar Horario Actividad
Elaborado por: Amalia Núñez

Una vez ingresada la información de horario en actividad, se procede a ingresar los indicadores de los beneficiarios de la actividad, dar click en el ícono de Indicadores.

Nota: Para poder ingresar los indicadores de los beneficiarios primero hay que seleccionar el horario en el que asistirán los beneficiarios.

Seleccionar 1	Charla	50	1	0,00	Charla sobre el VIH	Servicio Médico	Auditorio FISEI	Huachi Chico
Seleccionar 2	charla	100	0	0,00	Charla sobre el VIH	Servicio Médico	Auditorio FISEI	Huachi Chico
Seleccionar 3	Charla	20	0	0,00	Socializacion	Becas	Auditorio FISEI	Huachi Chico
Seleccionar 4	Estudio de la sexualidad	34	0	0,00	Charla sobre el VIH	Servicio Médico	Auditorio FISEI	Huachi Chico

HORARIOS

Código	Fecha	Hora Inicio	Hora Fin
Seleccionar 6	12/01/2016	0:00:00	10:10:00

Figura A.84: Pantalla Opción Indicadores
Elaborado por: Amalia Núñez

Pantalla de Ingreso Indicadores

Indicador	Descripción	Nº Participantes	Indicador	Descripción	Nº Participantes
Genero	Hombre:	23	Etnario	< 15 años:	20
	Mujer:	20		15 - 29 años:	0
	Total:	43		30 - 64 años:	23
Discapacidades	Física:	0		>64:	0
	Psicológica:	0		Total:	43
	Mental:	0	Indicador Descripción Nº Participantes	Indígenas:	0
	Auditiva:	0		Mestizos:	43
	Visual:	0		Bancos:	0
	Total:	0		Pueblos y Afroamericanos:	0
Movilidad	Ecuatorianos en el Extranjero:	0		Montubios:	0
	Extranjero en el Ecuador:	0		Otros:	0
	Total:	0	Total:	43	

Figura A.85: Pantalla de Ingreso Indicadores
Elaborado por: Amalia Núñez

Indicador	Descripción	Nº Participantes	Indicador	Descripción	Nº Participantes
Discapacidades	Física:	0	Indicador Descripción Nº Participantes	Indígenas:	0
	Psicológica:	0		Mestizos:	43
	Mental:	0		Bancos:	0
	Auditiva:	0		Pueblos y Afroamericanos:	0
	Visual:	0		Montubios:	0
	Total:	0		Otros:	0
Movilidad	Ecuatorianos en el Extranjero:	0	Total:	43	
	Extranjero en el Ecuador:	0			
	Total:	0			

Figura A.86: Pantalla de Ingreso Indicadores
Elaborado por: Amalia Núñez

Pantalla Calculo de Indicadores

En esta pantalla se puede ver los cálculos totales de los indicadores de los beneficiarios de las distintas Actividades.

Logo: UTA (Universidad Tecnológica del Azuay)

Título: DIBESAU DIRECCIÓN DE BIENESTAR ESTUDIANTIL Y ASISTENCIA UNIVERSITARIA PROGRAMAS DE SERVICIOS COMUNITARIOS

Menú de Navegación: Inicio, Responsables/Datos Laborales, Programas, Proyectos, Actividades, Informes, Administrar Usuarios

Submenú de Actividades: Lugares, Actividades/Proyecto, **Cálculo de Indicadores**, Problemas/Soluciones

Formulario: Buscar:

Indicador	Descripción	N° Participantes	Indicador	Descripción	N° Participantes
Genero	Hombre:	<input type="text"/>		< 15 años:	<input type="text"/>
	Mujer:	<input type="text"/>		15 - 29 años:	<input type="text"/>

Figura A.87: Pantalla Calculo de Indicadores
Elaborado por: Amalia Núñez

Pantalla Mostrar Actividades

Menú de Navegación: Inicio, Responsables/Datos Laborales, Programas, Proyectos, Actividades, Informes, Administrar Usuarios

Submenú de Actividades: Lugares, Actividades/Proyecto, Cálculo de Indicadores, **Mostrar Datos**

Formulario: Buscar:

Codigo	Actividad	Asi. Est.	Asi. Act.	Meta	Proyecto	Programa	Lugar	Dirección	Informe	Evidencia
Seleccionar 1	Charla	50	1	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 2	charla	100	0	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 3	Charla	20	0	0,00	Socializacion	Becas	Auditorio FISEI	Huachi Chico		
Seleccionar 4	Estudio de la sexualidad	34	0	0,00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		

Indicador	Descripción	N° Participantes	Indicador	Descripción	N° Participantes
Genero	Hombre:	<input type="text"/>		< 15 años:	<input type="text"/>
	Mujer:	<input type="text"/>		15 - 29 años:	<input type="text"/>

Figura A.88: Pantalla Mostrar Actividad
Elaborado por: Amalia Núñez

Nota: Para poder ver los resultados de los indicadores, seleccionar la actividad.

ACTIVIDADES

Buscar:

Codigo	Actividad	Asi. Est.	Asi. Act.	Meta	Proyecto	Programa	Lugar	Dirección	Informe	Evidencia
Seleccionar 1	Charla	50	1	0.00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 2	charla	100	0	0.00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 3	Charla	20	0	0.00	Socializacion	Becas	Auditorio FISEI	Huachi Chico		
Seleccionar 4	Estudio de la sexualidad	34	0	0.00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		

Figura A.89: Pantalla Seleccionar Actividad
Elaborado por: Amalia Núñez

Pantalla Mostrar Indicadores

INDICADORES

Indicador	Descripción	Nº Participantes
Genero	Hombre:	23
	Mujer:	20
	Total:	43
Discapacidades	Fisica:	0
	Psicologica:	0
	Mental:	0
	Auditiva:	0
	Visual:	0
	Total:	0
Etarario	< 15 años:	20
	15 - 29 años:	0
	30 - 64 años:	23
	>64:	0
	Total:	43
	Pueblos y Nacionalidades	Indigenas:
Mestizos:		43
Bancos:		0
Altoamericanos:		0
Montubios:		0
Otros:		0

Figura A.90: Pantalla Mostrar Indicadores
Elaborado por: Amalia Núñez

Pantalla Problemas Soluciones



DIBESAU
DIRECCIÓN DE BIENESTAR ESTUDIANTIL Y ASISTENCIA UNIVERSITARIA
PROGRAMAS DE SERVICIOS COMUNITARIOS

Bienvenido: **Administración!** | Cerrar Sesión

Inicio
Responsables/Datos Laborales
Programas
Proyectos
Actividades
Informes
Administrar Usuarios

ACTIVIDADES PR

Buscar:

- Lugares
- Actividades/Proyecto
- Cálculo de Indicadores
- Problemas/Soluciones

Figura A.91: Pantalla Problemas Soluciones
Elaborado por: Amalia Núñez

Pantalla Mostrar Actividades Proyecto

The screenshot shows the 'ACTIVIDADES PROYECTOS' interface. At the top, there is a navigation bar with links: Inicio, Responsables/Datos Laborales, Programas, Proyectos, Actividades, Informes, and Administrar Usuarios. Below the navigation bar is the title 'ACTIVIDADES PROYECTOS' and a search box labeled 'Buscar:'. The main content is a table with the following data:

Codigo	Actividad	Asi. Est.	Asi. Act.	Meta	Proyecto	Programa	Lugar	Dirección	Informe	Evidencia
Seleccionar 1	Charla	50	1	0.00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 2	charla	100	0	0.00	Charla sobre el VIH	Servicio Medico	Auditorio FISEI	Huachi Chico		
Seleccionar 3	Charla	20	0	0.00	Socializacion	Becas	Auditorio FISEI	Huachi Chico		
Seleccionar 4	Estudio de la	34	0	0.00	Charla sobre el Servicio		Auditorio	Huachi		

Below the table, there is a section titled 'PROBLEMAS' with four icons representing different problem types. Underneath, there is a small table with the following data:

Codigo	Nombre
Seleccionar 1	Poca participacion

At the bottom of the screenshot, there is a section titled 'SOLUCIONES'.

Figura A.92: Pantalla Mostrar Actividades Proyecto
Elaborado por: Amalia Núñez

Nota: Para poder ingresar los problemas primero hay que seleccionar la actividad a la que pertenece dicho problema.

Pantalla Ingreso Nuevo Problema

The screenshot shows the 'CREAR NUEVO PROBLEMA' form. At the top, there is a navigation bar with links: Inicio, Responsables/Datos Laborales, Programas, Proyectos, Actividades, Informes, and Administrar Usuarios. Below the navigation bar is the title 'CREAR NUEVO PROBLEMA'. The form contains the following fields:

- Actividad:
- Nombre:

Below the fields, there are two buttons: 'Cancelar' (with a red 'X' icon) and 'Guardar'.

Figura A.93: Pantalla Ingreso Nuevo Problema
Elaborado por: Amalia Núñez

Una vez ingresados los problemas procedemos a ingresar las soluciones.

Nota: Para poder ingresar las soluciones primero hay que seleccionar el problema al que corresponde dicha solución.

PROBLEMAS






	Codigo	Nombre
Seleccionar	1	Poca participacion
Seleccionar	3	Baja Autestima

SOLUCIONES

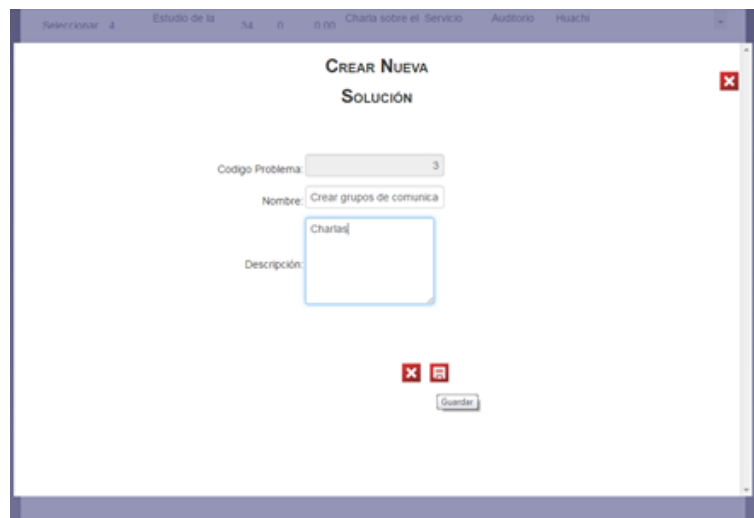





No existen resultados

Figura A.94: Pantalla Seleccionar Problema
Elaborado por: Amalia Núñez

Pantalla Ingreso Nueva Solución



CREAR NUEVA SOLUCIÓN

Codigo Problema:

Nombre:

Descripción:

Figura A.95: Pantalla Ingreso Nueva Solución
Elaborado por: Amalia Núñez

Pantalla Información Guardada Correctamente

PROBLEMAS

	Código	Nombre
Seleccionar	1	Poca participación
Seleccionar	3	Baja Atestima

SOLUCIONES

	Código Solución	Nombre	Descripción
Seleccionar	2	Crear grupos de comunicación	Charlas

Figura A.96: Pantalla Información Ingresada Correctamente
Elaborado por: Amalia Núñez

Pantalla Menú Informes

DIBESAU

DIRECCIÓN DE BIENESTAR ESTUDIANTIL Y ASISTENCIA UNIVERSITARIA

PROGRAMAS DE SERVICIOS COMUNITARIOS

Bienvenido: **Administración!** [Cerrar Sesión]

Inicio
Responsables/Datos Laborales
Programas
Proyectos
Actividades
Informes
Administrar Usuarios

- DIBESAU
- BECAS
- UTA
- CALENDARIO

- Planificación Programas
- General Programas
- Actividades Ejecutadas
- Actividades No Ejecutadas
- Proyecto por Fechas

enero de 2015

lun	mar	mié	jue	vie	sáb	dom
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figura A.97: Pantalla Menú Informes
Elaborado por: Amalia Núñez

Pantalla Informes Opción Planificación Programas



Figura A.98: Pantalla Menú Informes
Elaborado por: Amalia Núñez

Pantalla Informes Planificación Programas



Figura A.99: Pantalla Menú Informes
Elaborado por: Amalia Núñez

1. Título.
2. Guardar informe en los formatos de Excel, Pdf, Word.
3. Actualizar informe.
4. Encabezado informe.
5. Cuerpo del informe.

Nota: El mismo procedimiento se lo realiza para acceder a todos los informes.

Anexo B

Guía de Observación

Universidad Técnica de Ambato

Facultad de Ingeniería en Sistemas, Electrónica e Industrial



Guía de Observación de procesos para gestión de programas

Estudio de campo mediante el método de observación en el DIBESA de la UTA.

Guía de Observación	
Objetivo	Obtener información sobre de los procesos que se realizan para la gestión de programas de servicios comunitarios.
Tipo de Observación	Directa.
Indicadores a Observar	
Medios para recolección de la información	Formato de plan operativo anual 2015 para gestión de los programas comunitarios.
Medios en que se almacena la información generada	Carpetas físicas. Muebles archivadores. Cajas de cartón.
Disponibilidad de la información	Búsqueda de archivos físicos en carpetas.
Análisis	Se pudo observar un formato del plan operativo anual realizado por un profesional del DIBESAU, en el cual se evidencio los principales procesos para la gestión de programas como, los proyectos, las actividades, las metas, el número de beneficiarios y el periodo. Lo cual ayudó a obtener una visión de lo que se realiza para la gestión de programas.

Tabla B.1: Guía de Observación

Anexo C

Entrevistas

Universidad Técnica de Ambato

Facultad de Ingeniería en Sistemas, Electrónica e Industrial



Entrevista dirigida a los responsables de la gestión de programas de servicios comunitarios del DIBESAU de la UTA.

Entrevistado(a): Dra. Eulalia Pino Loza Directora DIBESAU de la UTA.

Entrevistadora: Egresada Amalia Núñez. C.I. 1804600748.

Lugar: Campus Huachi de la UTA, Dirección de Bienestar Estudiantil y Asistencia Universitaria (DIBESAU).

Fecha: 28 de marzo de 2015.

Objetivo: Obtener información de los distintos procesos que se realizan en la gestión de programas de servicios comunitarios, así también los problemas que se generan al no contar con un sistema para gestión de programas de servicios comunitarios, para mejorar el mismo mediante el desarrollo de una aplicación Web.

Indicaciones: Contestar las siguientes preguntas con absoluta sinceridad y claramente en base a su criterio y experiencia.

1. ¿Actualmente cómo se lleva la gestión de los programas?

En los programas se realizan varios proyectos. Los profesionales encargados de los proyectos, realizan un informe manual sobre los programas, proyectos, todas las actividades, los objetivos, las metas y el número de beneficiarios.

2. ¿Quién o quiénes son los responsables de los proyectos?

Los proyectos tienen varias responsables como, la educadora, los médicos, psicólogos.

3. ¿Quién o quiénes son los responsables de los programas?

Los programas son responsabilidad de la dirección o de un encargado.

4. Detallar los procesos para la gestión de un programa.

Un programa tiene varios proyectos que cumplir.

5. Detallar los procesos para la gestión de un proyecto.

Un proyecto tiene un objetivo y varias actividades por cumplir.

6. Detallar los procesos para la gestión de las actividades.

Cada actividad tiene metas, horarios y beneficiarios.

7. ¿Qué problemas ha evidenciado en el manejo de la información de los programas?

Muchos problemas debido a que todo se lo hace manualmente, los profesionales del DIBESAU realizan los informes a su criterio, por lo que es complicado sacar conclusiones de los programas oportunamente, tenemos que analizar cada informe para poder sacar las conclusiones deseadas.

Universidad Técnica de Ambato

Facultad de Ingeniería en Sistemas, Electrónica e Industrial



Entrevista dirigida a los responsables de la gestión de programas de servicios comunitarios del DIBESAU de la UTA.

Entrevistado(a): Miguel López Técnico DIBESAU.

Entrevistadora: Egresada Amalia Núñez. C.I. 1804600748.

Lugar: Campus Huachi de la UTA, Dirección de Bienestar Estudiantil y Asistencia Universitaria (DIBESAU).

Fecha: 28 de marzo de 2015.

Objetivo: Obtener información de los distintos procesos que se realizan en la gestión de programas de servicios comunitarios, así también los problemas que se generan al no contar con un sistema para gestión de programas de servicios comunitarios, para mejorar el mismo mediante el desarrollo de una aplicación Web.

Indicaciones: Contestar las siguientes preguntas con absoluta sinceridad y claramente en base a su criterio y experiencia.

1. **¿Actualmente como se manejan los horarios para las actividades?**

Se maneja de forma manual dependiendo de la planificación de cada responsable.

2. **¿Cual es el proceso para el registro de beneficiarios?**

El registro es por genero cuantos hay, por discapacidades, por movilidad, por etario y pueblos y nacionalidades

3. **¿Cual es el proceso para saber si una actividad fue realizada o no?**

Actualmente no se tiene esa información, pero es sumamente importante, seria si las actividades tuvieron beneficiarios fueron ejecutadas, caso contrario no.

4. ¿Qué problemas ha evidenciado en el manejo de la información de las actividades?

Varios problemas, debido a que no se cuenta con un sistema para gestión de todo estos procesos, en muchas ocasiones la información es incompleta, las metas no se cumplen.

5. ¿Cuáles son los parámetros para que las metas se cumplan ?

Se toma a consideración el número de asistentes programados y el número total de beneficiarios, según la diferencia entre los dos valores sacamos el porcentaje de cumplimiento de la meta.

6. ¿Qué información adicional desearía tener?

Actualmente en los informes no se registran los problemas detectados en las actividades, lo cual es muy necesario para poder darles solución y tener una idea más profunda de los problemas que se pueden encontrar en cada una de las actividades.