



3.- Desarrollo e implementación del sistema.

3.1.- Funcionalidad de entrar al sistema

Para poder acceder al sistema el alumno tendrá que escribir su nombre de usuario y su password, a continuación se muestra el diseño, ver figura 3.1.

The screenshot shows a web-based login interface. At the top, there is a blue header bar with the text "Sistema para el control de asistencia de alumnos del servicio social." on the left and right sides of the header are two logos: the Faculty of Engineering logo on the left and the Faculty of Social Services logo on the right. Below the header is a light green background. In the center, there is a login form with two input fields: "Usuario" and "Password". Below the "Password" field are two buttons: "Entrar" and "Salir". Below the buttons is a link that says "Ir a administración".

Figura 3.1 Logeo para acceder al sistema.

Esta interfaz cuenta con varias validaciones por ejemplo:

- Si el alumno no se encuentra registrado no lo dejará acceder al sistema.
- Si la dirección IP de la máquina, no coincide con la dirección ip que se le asignó desde un principio, significa que no es su máquina y no lo dejará acceder.
- Asimismo las cajas de texto están validadas, por ejemplo en la caja de texto para usuario solo se aceptan caracteres y en password solo números.



3.1.1.- Funcionalidades al entrar al sistema

En esta parte se desarrollaron las funcionalidades para el control de asistencia del alumno, entre las cuales están: Registrar Entrada, Registrar Salida, Consulta y Generar Reporte bimestral. A continuación se muestra una imagen de estas funcionalidades, ver figura 3.2.



Figura 3.2 Registro de entrada y salida.

El botón “Registrar Entrada”:

Registra la entrada de entrada del alumno.

El botón “Registrar Salida”:

Registrar la salida del alumno

El botón “Consulta”:

Muestra el registro de horas del alumno que lleva hasta el momento, en este caso se muestra el número de cuenta, hora entrada, hora salida, fecha y el número total de horas hasta ese momento. Ver figura 5.3.



Consulta de horas

[Regresar](#)

Consulta de información			
Número Cuenta	Entrada	Salida	Horas
22	20/08/2010 11:19:00 p.m.	20/08/2010 11:59:00 p.m.	0.666666666666667
22	25/08/2010 09:36:00 a.m.	25/08/2010 05:10:00 p.m.	7.566666666666667
22	26/08/2010 09:36:00 a.m.	26/08/2010 12:36:00 p.m.	3
22	27/08/2010 09:36:00 a.m.	27/08/2010 11:36:00 a.m.	2
22	30/08/2010 09:08:00 a.m.		
22	01/09/2010 08:25:00 p.m.	01/09/2010 11:25:00 p.m.	3
22	02/09/2010 06:57:00 p.m.	02/09/2010 06:58:00 p.m.	0.016666666666667
22	06/09/2010 08:46:00 p.m.		
22	26/10/2010 10:56:00 p.m.	26/10/2010 11:10:00 p.m.	0.233333333333333
22	05/11/2010 09:54:00 p.m.		
22	06/11/2010 03:48:00 p.m.		
22	07/11/2010 03:07:00 p.m.	07/11/2010 04:09:00 p.m.	1.033333333333333
22	09/11/2010 10:16:00 p.m.		
22	01/12/2010 10:44:00 p.m.		
22	01/12/2010 10:45:00 p.m.	01/12/2010 10:46:00 p.m.	0.016666666666667
22	14/12/2010 11:11:00 p.m.		
22	15/12/2010 09:25:00 p.m.		
22	16/12/2010 09:56:00 p.m.		

Total de horas: 17.5333333333333

Figura 3.3 Consulta de horas



3.1.2.- Generar Reporte Bimestral

En esta funcionalidad se desarrolló una interfaz para poder generar el reporte bimestral de las actividades realizadas en el periodo correspondiente, en esta parte se realiza una captura de datos por parte del alumno. Ver figura 3.4

Captura de reporte bimestral	
Numero de registro	<input type="text"/>
Numero de Informe	<input type="text"/>
Periodo de inicio	19/12/2010 <input type="button" value="..."/>
Periodo de termino	19/12/2010 <input type="button" value="..."/>
Nombre de la dependencia	<input type="text"/>
Nombre del programa	APOYO AL LABORATORIO DE CÓMPUTO DE ALTO DESEMPEÑO <input type="button" value="v"/>
Clave DGOSE	<input type="text"/> - <input type="text"/> / <input type="text"/> - <input type="text"/>
Actividad1	<input type="text"/> <input type="button" value="v"/> horas <input type="text"/>
Actividad2	<input type="text"/> <input type="button" value="v"/> horas <input type="text"/>
Horas en el Bimestre	<input type="text"/>
Horas Acumuladas	<input type="text"/>
Resultados a sociedad	<input type="text"/>
Resultados formacion profesional	<input type="text"/>
<input type="button" value="Limpiar"/> <input type="button" value="Enviar"/>	

Figura 3.4 Captura de reporte bimestral

Esta interfaz cuenta con varias validaciones por ejemplo:

- El número de registro solo acepta números.
- El número de informes solo acepta caracteres
- Debe ingresar al menos una actividad.
- Se debe ingresar los resultados a la sociedad y resultados en formación profesional.

En un inicio el alumno podrá agregar tres actividades con sus respectivas horas, pero si el alumno lo desea podrá agregar hasta tres actividades más, pudiendo tener un total de 6 actividades.



3.2.- Funcionalidad de administración

En esta parte del sistema se llevará a cabo la administración de los alumnos, la cual sólo podrá tener acceso la persona a cargo de la administración del sistema. Para poder acceder a la parte de la administración se tendrá que ingresar nombre de usuario y password. Ver figura 3.5

The screenshot shows a web interface for administrator login. At the top, there is a blue header bar containing the text "Logeo de administrador" in the center, flanked by two logos: the "INGENIERIA" logo on the left and the institutional coat of arms on the right. Below the header is a large olive-green area. In the center of this area is a login form with two input fields labeled "Administrador" and "Password" in red text. Below these fields are two buttons: "Entrar" and "Salir". At the bottom of the form area is a blue underlined link that says "Ir a inicio".

Figura 3.5 Logeo de administrador



Una vez dentro del sistema se podrá acceder a las siguientes funcionalidades.
Ver figura 3.6



Figura 3.6 Administración

Aquí se tendrá todo lo relacionado con respecto al alumno en cuestión de altas, bajas, modificaciones y generar carta de inicio y de término. También se podrá modificar el nombre de los coordinadores.



3.2.1.-Ingresar alumno

Se diseñó una interfaz para capturar los datos del alumno y así darlos de alta. Figura 3.7

The screenshot shows a web form titled "Datos de alumno a ingresar" (Student Data to be entered). The form is set against a blue header with the faculty logo on the left and right. The form fields are as follows:

Nombre	<input type="text"/>
ApellidoPaterno	<input type="text"/>
ApellidoMaterno	<input type="text"/>
Telefono	<input type="text"/>
CorreoElectronico	<input type="text"/>
Carrera	Ingeniería en Computación <input type="button" value="v"/>
FechaInicio	26/02/2011 <input type="button" value="..."/>
Horario	Lunes a viernes 2 horas <input type="button" value="v"/>
Programa	APOYO AL LABORATORIO DE CÓMPUTO DE ALTO DESEMPEÑO <input type="button" value="v"/>
NoCuenta	<input type="text"/>
IP	<input type="text"/>

At the bottom of the form are two buttons: "Regresar" and "enviar".

Figura 3.7 Ingresar alumno

Esta interfaz cuenta con varias validaciones por ejemplo:

- El nombre solo acepta caracteres, al igual que el campo de apellido paterno y materno.
- El teléfono solo acepta números y de una longitud de 8 caracteres.
- Número de cuenta solo acepta números.



3.2.2.- Editar alumno

Se ingresará el número de cuenta del alumno para poder editar sus datos. Figura 3.8.

The screenshot shows a web page with a blue header. On the left of the header is the 'INGENIERIA' logo, and on the right is the golden emblem. The title 'Actualizar alumno' is centered in the header. Below the header is a light green background. In the center, there is a search form with a text input field, a 'Regresar' button, and an 'enviar' button.

Figura 3.8 Actualizar Alumno

Una vez enviado el número de cuenta se mostrará la siguiente interfaz para modificar los datos del alumno, ver Figura 3.9.



Actualizar alumno

Nombre	Thomas
ApellidoPaterno	Gonzales
ApellidoMaterno	Torres
Telefono	56787656
CorreoElectronico	Torres_56@yahoo.com.mx
Carrera	Ingeniería en Telecomunicación
FechaInicio	26/02/2011
Horario	Lunes a viernes 4 horas
Programa	APOYO AL LABORATORIO MICROSOFT RESEARCH
NoCuenta	25 <input type="checkbox"/> Cambiar número de cuenta
IP	121.0.0.2

Figura 3.9 Editar alumno.

3.2.3.- Eliminar

Se ingresará el número de cuenta del alumno para poder darlo de baja. Figura 3.10

Eliminar alumno

Ingresa número de cuenta de alumno:

Figura 3.10 Eliminar alumno



Una vez dando click en el botón de enviar, se eliminará el alumno y se mostrará la ventana de confirmación. Figura 3.11.



Figura 3.11 Ventana de confirmación



3.2.4.- Funcionalidad Generar reportes.

Esta parte fue diseñada para poder generar la carta de inicio y de término del alumno del servicio social. Para desarrollar esta parte me apoyé de Interop, el cual es un framework de .NET para poder insertar texto desde código en c# a word. También tuve que crear unas plantillas en Word a las cuales les agregue content controls para poder insertar el texto desde c#.

Para poder generar la respectiva carta se tendrá que ingresar el número de cuenta del alumno. Figura 3.12

Generar Cartas

Ingrese el número de cuenta del alumno para generar la carta

NoCuenta

[Regresar a administración](#)

Figura 3.12 Funcionalidad Generar Reporte.



3.2.5.- Funcionalidad modificar coordinadores

Esta funcionalidad se encarga de actualizar el nombre del director de la facultad de ingeniería, coordinador de la carrera de ingeniería en computación y coordinador del servicio social, esto en caso de que cambie de nombre.

Actualizar coordinadores		
Director	M. EN A. JOSÉ GONZALO GUERRERO ZEPEDA	<input type="radio"/> Actualizar
Coordinador	M. I. Jorge Valeriano	<input type="radio"/> Actualizar
CoordinadorSS	Lic. Angélica Gutiérrez Vázquez	<input type="radio"/> Actualizar
<input type="button" value="Regresar"/>		<input type="button" value="Actualizar"/>

3.13 Funcionalidad modificar coordinadores.



3.2.6. Funcionalidad revisar horas del alumno

Esta funcionalidad permitirá elegir un alumno, que se encuentre registrado en la base de datos, para revisar las horas acumuladas hasta ese momento.

Elige el alumno a revisar horas

Elige el alumno: almaraz flores bere

Regresar Aceptar

3.14 Revisar horas del alumno.



Una vez dentro de la funcionalidad el administrador podrá ver las horas del alumno. También podrá como modificar alguna hora de entrada o salida.

Consulta de información				
No. Registro	Número Cuenta	Entrada	Salida	Horas
38	211007	13/01/2010 02:50:00 a.m.	13/01/2010 05:37:00 a.m.	2.78
36	211007	22/01/2011 03:30:00 p.m.	22/01/2011 08:30:00 p.m.	5
42	211007	30/01/2011 01:08:00 a.m.	30/01/2011 01:12:00 a.m.	0.07
Mes: Enero Horas: 7.85				
48	211007	26/02/2011 02:52:00 p.m.		

Horas total: 7.85

[Regresar](#)

Ingresa los siguientes datos para actualizar una fecha.	
Número Registro	<input type="text"/>
Entrada	<input type="text"/> : <input type="text"/> Hrs:Min
Salida	<input type="text"/> : <input type="text"/> Hrs:Min
Aceptar	

3.15 Revisar horas del alumno y modificación de alguna hora de entrada o salida.



A continuación se puede ver una parte del diseño de la carta de inicio con sus respectivos content controls los cuáles podemos reconocer porque son etiquetas de color azul. Ver Figura 3.16

México D. F. a de de

M. EN A. JOSE GONZALO GUERRERO ZEPEDA
Director de la Facultad de
Ingeniería de la UNAM
Presente:

Asunto: Carta de aceptación para la prestación
del Servicio Social del C.

Me permito informar a usted nuestra conformidad para que el C. con número de cuenta de la carrera que se imparte en la Facultad a su digno cargo, preste su Servicio Social en esta Dependencia durante un período de **seis** meses, a partir del de de colaborando **20** horas a la semana, con horario de en el programa de trabajo número desarrollando las siguientes actividades fundamentales:

1. Programación de cualquier lenguaje.
2. Manejo de bases de datos.
3. Mantenimiento y administración de equipos de cómputo.
4. Soporte a usuarios.
5. Mantener abierto y funcionando el laboratorio de Microsoft.
6. Participación en grupos de trabajo para el diseño y desarrollo de proyectos.

Figura 3.16 Diseño de la carta de inicio del servicio social



3.3.- Clases del sistema

El sistema cuenta con cuatro clases para su funcionamiento, las cuales se encuentra en código en c#, a continuación se muestra una breve descripción de las clases y sus respectivo código.

3.3.1.- Clase alumno

En la clase alumno se van a encontrar todos los datos del alumno, esto es sus atributos, para poder realizar la programación orientada a objetos. A continuación se muestra su código.

```
public class Alumno
{
    private string _Nombre;
    private string _ApellidoMaterno;
    private string _ApellidoPaterno;
    private string _Telefono;
    private string _CorreoElectronico;
    private DateTime _FechaInicio;
    private DateTime _FechaTermino;
    private Int32 _IdCarrera;
    private Int32 _IdHorario;
    private Int32 _IdPrograma;
    private Int32 _NoCuenta;
    private Int32 _NuevoNoCuenta;
    private string _IP;

    #region Propiedades

    public string Nombre
    {
        get { return this._Nombre; }
        set { this._Nombre = value; }
    }

    public string ApellidoMaterno
    {
        get { return this._ApellidoMaterno; }
        set { this._ApellidoMaterno = value; }
    }

    public string ApellidoPaterno
    {
        get { return this._ApellidoPaterno; }
        set { this._ApellidoPaterno = value; }
    }
}
```




```
public string Telefono
{
    get { return this._Telefono; }
    set { this._Telefono = value; }
}

public string CorreoElectronico
{
    get { return this._CorreoElectronico; }
    set { this._CorreoElectronico = value; }
}

public DateTime FechaInicio
{
    get { return this._FechaInicio; }
    set { this._FechaInicio = value; }
}

public DateTime FechaTermino
{
    get { return this._FechaTermino; }
    set { this._FechaTermino = value; }
}

public Int32 NoCuenta
{
    get { return this._NoCuenta; }
    set { this._NoCuenta = value; }
}

public Int32 NuevoNoCuenta
{
    get { return this._NuevoNoCuenta; }
    set { this._NuevoNoCuenta = value; }
}

public Int32 IdCarrera
{
    get { return this._IdCarrera; }
    set { this._IdCarrera = value; }
}

public Int32 IdHorario
{
    get { return this._IdHorario; }
    set { this._IdHorario = value; }
}

public Int32 IdPrograma
{
    get { return this._IdPrograma; }
    set { this._IdPrograma = value; }
}
```



```
public string IP
{
    get { return this._IP; }
    set { this._IP = value; }
}

#endregion

public Alumno()
{
}
}
}
```



3.3.2.- Clase ConexionBD

Esta clase se encarga de abrir y cerrar una conexión a la base de datos, recibe en su constructor la cadena de conexión.

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

/*
 * Clase para conectarme a la base de datos
 * Descripción: esta clase me permite crear un objeto para conectarme .
 */

public class ConexionBD
{
    //Atributos
    private string cadenaConexion="";
    public SqlConnection conexion;

    //Creo mi objeto para la conexion, se le manda la cadena de
    conexion.
    public ConexionBD(string cadenaConexion)
    {
        this.conexion = new SqlConnection(cadenaConexion);
    }

    //Metodo para abrir la conexion y la regreso.
    public SqlConnection obtieneConexion()
    {
        this.conexion.Open();
        return this.conexion;
    }

    //Se cierra la conexion.
    public void cerrarConexion()
    {
        conexion.Close();
    }
}
```



3.3.3.- Clase de AlumnoNegocio.

Esta clase se encarga de la parte de negocio del sistema, va a tener un objeto de la clase ConexionBD para poder conectarse a la base de datos. Aquí solo muestra una parte del código ya que debido que contiene todos los métodos del sistema es muy grande.

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Net;
using System.Net.NetworkInformation;

/// <summary>
/// AlumnoNegocio se encarga de la parte de negocio del sistema,
/// Esta clase tiene un objeto ConexionBD para obtener una conexion a la Base
Datos
/// </summary>
public class AlumnoNegocio
{
    private ConexionBD conexion = null;
    Alumno alumno = null;

    public AlumnoNegocio()
    {
        conexion = new
ConexionBD(ConfigurationManager.ConnectionStrings["conexion"].ConnectionString);
    }

    /// <summary>
    /// Busca un alumno por su numero de cuenta.
    /// </summary>
    /// <param name="NoCuenta">NoCuenta de alumno</param>
    /// <returns>Regresa el alumno</returns>
    public Alumno buscarAlumno(Int32 NoCuenta)
    {
        this.alumno = new Alumno();
        SqlCommand comando = new SqlCommand("spBuscaAlumnoXcuenta",
conexion.obtieneConexion());
        comando.CommandType = CommandType.StoredProcedure;
        comando.Parameters.AddWithValue("@NoCuenta", NoCuenta);
        SqlDataReader reader = comando.ExecuteReader();

        while(reader.Read())
        {

            alumno.Nombre = (String)reader["Nombre"].ToString();
        }
    }
}
```



```
        alumno.ApellidoPaterno =
(String)reader["ApellidoPaterno"].ToString();
        alumno.ApellidoMaterno =
(String)reader["ApellidoMaterno"].ToString();
        alumno.NoCuenta = Convert.ToInt32(reader["NoCuenta"]);
        alumno.IdCarrera = Convert.ToInt32(reader["IdCarrera"]);
    }

    reader.Close();
    conexion.cerrarConexion();

    return alumno;
}

public DataTable cargaCatalogo(string NombreSP)
{
    SqlDataAdapter adaptador = new SqlDataAdapter(NombreSP,
conexion.obtieneConexion());
    DataTable tabla = new DataTable();
    adaptador.Fill(tabla);

    conexion.cerrarConexion();

    return tabla;
}

public DataTable cargaVistaAlumno(string NombreSP, Int32 NoCuenta)
{
    SqlDataAdapter adaptador = new SqlDataAdapter();
    SqlCommand comando = new SqlCommand(NombreSP,
conexion.obtieneConexion());
    comando.CommandType = CommandType.StoredProcedure;
    adaptador.SelectCommand = comando;
    comando.Parameters.Add("@NoCuenta", SqlDbType.Int).Value = NoCuenta;

    DataTable tabla = new DataTable();
    adaptador.Fill(tabla);

    conexion.cerrarConexion();

    return tabla;
}

public String regresaCarrera(Int32 NoCueta)
{
    SqlCommand comando = new SqlCommand("spSelectCarreraByIdCarrera",
conexion.obtieneConexion());
    comando.CommandType = CommandType.StoredProcedure;
    comando.Parameters.AddWithValue("@NoCuenta", NoCueta);

    String carrera = (String)comando.ExecuteScalar();
    conexion.cerrarConexion();

    return carrera;
}
```



```
public String obtenerIP()
{
    IPEndPoint host;
    string localIP = "";
    host = Dns.GetHostEntry(Dns.GetHostName());

    foreach (IPAddress ip in host.AddressList)
    {
        if (ip.AddressFamily.ToString() == "InterNetwork")
        {
            localIP = ip.ToString();
        }
    }

    return localIP;
}

public void insertafechaTermino(Int32 NoCuenta, DateTime fechaTermino)
{
    SqlCommand comando = new SqlCommand("spUpdAlumnoFechaTermino",
conexion.obtieneConexion());
    comando.CommandType = CommandType.StoredProcedure;
    comando.Parameters.Add("@NoCuenta", SqlDbType.Int).Value = NoCuenta;
    comando.Parameters.Add("@fechaTermino", SqlDbType.DateTime).Value =
fechaTermino;

    comando.ExecuteNonQuery();
}

public void cerrarConexion()
{
    conexion.cerrarConexion();
}
```

3.3.4.- Clase GenerarCartas

Esta clase se encargará de generar la carta de inicio y de término del alumno así como su reporte bimestral. Esta clase recibe en su constructor la ruta de la plantilla, los datos del alumno, ruta destino donde se guardará y el tipo de carta a generar.