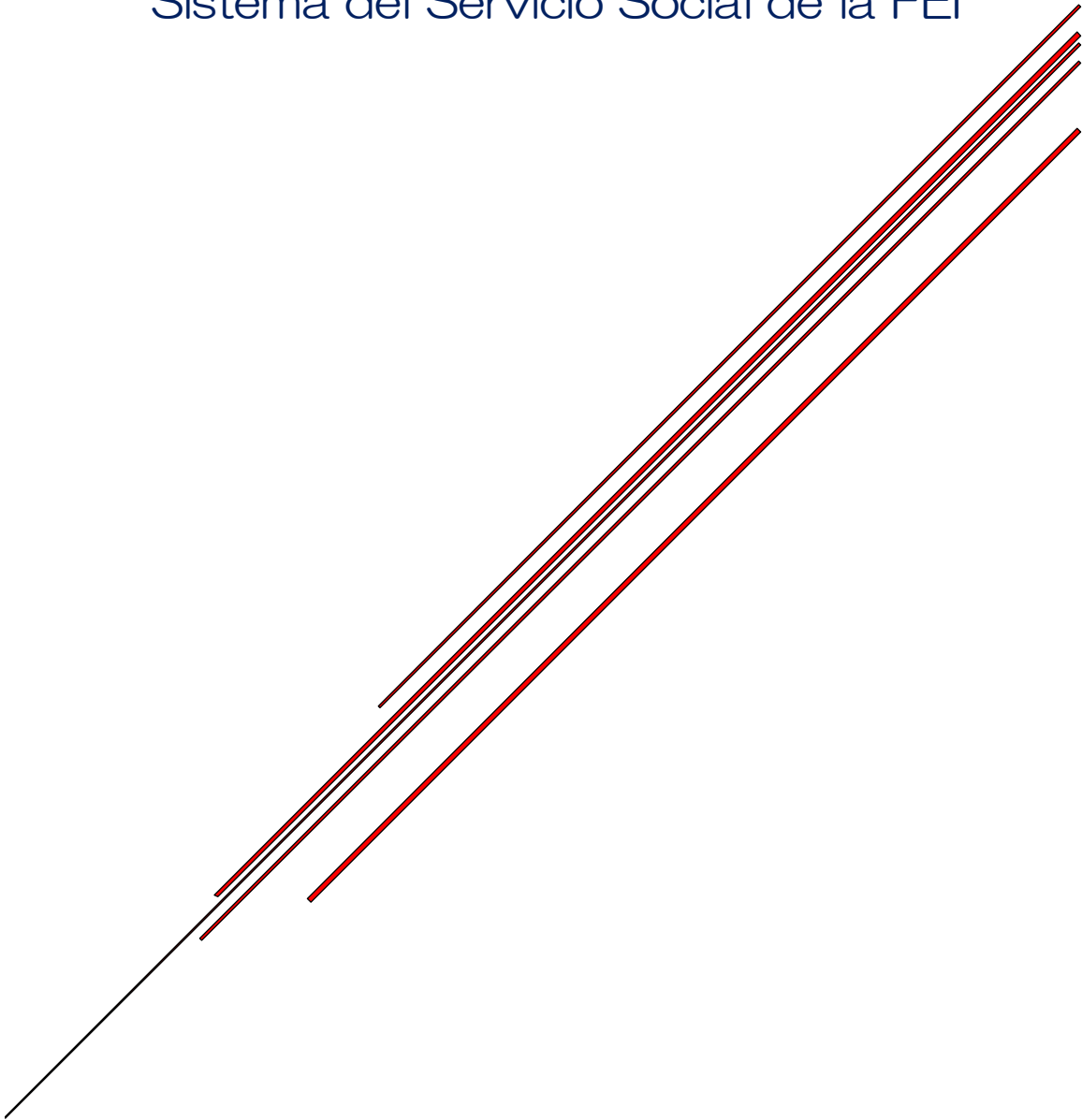


PROYECTO FINAL

Sistema del Servicio Social de la FEI



Equipo 5: Dulce Regina Saavedra Mata, Arturo Villa López,
Principios de Construcción

Estándar de Código

1. Introducción

1.1. Objetivo

En el siguiente documento se establecen algunas normas o reglas que ayudan a la construcción de software para brindar una mejor interpretación.

- Se asegura la calidad del producto de software.
- Facilita el mantenimiento de este.
- Simplifica la detección de errores y bugs.
- Facilitar la comprensión para el trabajo en conjunto.

Todos estos conceptos van alineados con el uso de la Metodología Orientada a Objetos en la plataforma Visual Studio.NET

1.2. Alcance

Para este proyecto en específico desarrollador en el lenguaje C# en Visual Studio. Incluye los estándares de nomenclatura, diseño del código, uso de comentarios, así como unas buenas prácticas.

2. Nomenclaturas

Existen varios estilos de estándares para definir los nombres e identificar los elementos del proyecto en C#, entre los cuales destacan 2:

Notación de Pascal (Pascal Case): El primer carácter de todas las palabras se escriben en Mayúscula y todos los demás caracteres en minúsculas. Por ejemplo: CódigoPrueba.

Notación de camello (Camel Case): El primer carácter de todas las palabras excepto de la primera se escriben con Mayúscula y todos los demás en minúsculas. Por ejemplo: códigoPrueba.

2.1. Variables

- Los nombres de variables y de atributos de clase deberán de ser de más de un carácter.

Una excepción pueden ser los nombres de los iteradores en los ciclos.

Ejemplo:

```
For (int i = 0 ; i < Variable ; i ++ ) {  
    }  
}
```

- Usar notación Camel Case.
- Distinguir los nombres de las variables booleanas.
- Las instancias usan la notación Camel Case
- Diferenciar entre variables globales y locales.
- No usar palabras reservadas en nombres de variables.
- Las variables booleanas de retorno se escriben con Camel case.

Ejemplo:

Correcto:

```
string Nombre;  
int NumeroDeAlumnos;
```

Incorrecto:

```
string nom;  
int nA;
```

2.2. Constantes

La definición de los constantes se efectúa siempre y cuando el valor no cambie durante el tiempo de ejecución del software, todo en mayúsculas.

Ejemplo:

```
public const int MESES = 12;
```

2.3. Métodos

La definición del nombre de los métodos este guiado por las siguientes normas:

- Utilizar Pascal Case.
- Los metodos de prueba unitaria se inicializan con la palabra “Prueba” seguido del tipo de prueba.

Ejemplo : PruebaRegistrarAlumnoValido().

- Utilizar verbos o frases descriptivas sobre la acción que tendrá en el programa.
- Se usan prefijos para nombrar métodos y clases.
- No se deben usar nombres extraños o engañosos, el nombre debe ser obvio por lo cual no se necesita documentación para explicarlo.

Por Ejemplo: VaciarCubeta(), TirarPelota(), CerrarVentana().

2.4. Clases

La definición del nombre de las clases este guiado por las siguientes reglas:

- Se debe usar un sustantivo en singular, con excepción que la clase represente múltiples cosas.
- Se debe utilizar Pascal Case.
- No se deben utilizar prefijos como C, T o clr.
- Las clases solo representas “cosas” y no “acciones”.

Ejemplo: Empleado, Proveedor, CuentaBancaria.

2.5. Interfaces

Usa el prefijo “I” con notación **Pascal** para las interfaces.

//Ejemplo:

```
interface IClaseDAO{  
    void Metodo1(int variable);  
    int Metodo2();  
}
```

2.6. Interfaces Gráficas

La definición del nombre de las interfaces este guiado por las siguientes reglas:

- Se debe utilizar pascal case.

- La nomenclatura es “Actor que usa la pantalla” + “Elemento o Actor con el que interactúa”.

2.7. Objetos y Listas de Objetos

La definición del nombre de las interfaces este guiado por las siguientes reglas:

- Se escriben con letra mayúscula al inicio.
- La nomenclatura es “Actor que usa la pantalla” + “Elemento o Actor con el que interactúa”.

3. Diseño de código

3.1. Estructura

Estas convenciones tienen la finalidad siguiente:

- Facilitar la lectura, modificación y mantenimiento del código.
- Dar un aspecto coherente al código, de tal manera que el lector centre su atención en el contenido y no en el diseño.
- Permitir la lectura y comprensión con mayor rapidez.
- Utilizar una sola instrucción por línea.
- Utilizar una sola declaración por línea.
- Si a las líneas no se les aplica una sangría automática, aplicarla con una tabulación (cuatro espacios).
- Utilizar un espacio en blanco para separar grupos lógicos de códigos.
- Utilizar paréntesis para cláusulas de una expresión, así se hace más evidente.

Ejemplo:

```
if ((numero1 < numero2) && (numero1 < numero3))
{
    //Ingresar código
}
```

4. Uso de Comentarios

Los comentarios deberán seguir las siguientes convenciones:

- No usar comentarios para cada línea y/o variable.
- Usar de preferencia `//` en lugar de `/**/`
- Usar los menos comentarios posibles.
- Usarlos para explicar lógica compleja.
- Usarlos para explicar asignaciones a las variables.
- Tener buena ortografía, puntuación y gramática.

5. Documentación de la API

C# utiliza XML para generar la API del proyecto.

- Los comentarios utilizados para la documentación de clases y métodos usan tres barras diagonales: `///` y un cuerpo de comentario con formato XML. Por ejemplo:

```
/// <summary>
/// Esta clase tiene una funcion.
/// </summary>
public class claseRandom
{
}
```

La base para la documentación de una clase cualquiera en este proyecto es:

```
/// <summary>
/// Explicación de la clase.
/// </summary>
public class claseRandom
{
}
```

La base para la documentación de un método en este proyecto es:

```
/// <summary>
```

```

/// Explicación del metodo.

/// </summary>

/// <param name="nombre de parametro">Tipo de parametro.</param>

/// <returns>

/// Aclaración de valor que regresa.

/// </returns>

/// <exception cref="excepción">Condición para aparición de la excepción.</exception>

private void saludo(String mensaje)
{
}

```

5.1. Etiquetas

Las etiquetas que se pueden realizar están redactadas en la web .NET de Microsoft.

<summary>

Requerido

Se utiliza usualmente para describir un objeto. Es la primera línea de las etiquetas de la documentación.

<seealso>

Opcional

Agrega enlaces asociados al método.

<param>

Requerido

Describe un parámetro en un método.

<typeparam>

Requerido

Describe un parámetro o método de tipo genérico.

<returns>

Requerido

Se utiliza para describir el valor a retornar por un método.

<exception>

Opcional

Describe una excepción que podría ser lanzada por un método.

<permission>

Opcional

Se utiliza para describir el nivel de acceso a un método.

<include>

Opcional

Utilizado para incluir en la documentación un archivo o parte de otro en el código.

<para>

Opcional

Utilizado para definir un bloque, solo se utiliza a nivel del elemento de documentación.

<code>

Opcional

Se utiliza para especificar el formato que tendrá el código.

<see>

Opcional

Se utiliza para definir un enlace a una página interna (cref) o externa (href).

<paramref>

Opcional

Hace referencia a un parámetro dentro de otra etiqueta

<typeparamref>

Opcional

Hace referencia a un tipo de parámetro genérico

6. Buenas prácticas

1. Evitar escribir métodos muy largos. Se recomienda que un método deba tener entre 1 a 25 líneas de código, si se llega a tener más de 25 líneas se deberá refactorizar , las excepciones no hacen que el método sea mas largo.


```

public bool registrarAlumno(Alumno alumno)
{
    bool alumnoGuardado = false;
    ConexionBaseDatos conexionBaseDatos = new ConexionBaseDatos();
    using (SqlConnection connection = conexionBaseDatos.GetConnection())
    {
        connection.Open();
        using (SqlCommand command = new SqlCommand("Insert into Alumno_SS(Matricula_Alumno," +
            " Nombre_Alumno, Apellidos_Alumno, Correo_Alumno, FechaNacimiento, Contraseña_Alumno, " +
            "Num_Horas, Carrera) values(@matricula, @nombre, @apellido, @correo, @fechanacimiento, @contraseña," +
            " @numhoras, @carrera)", connection))
        {
            command.Parameters.Add(new SqlParameter("matricula", alumno.Matricula));
            command.Parameters.Add(new SqlParameter("nombre", alumno.Nombre));
            command.Parameters.Add(new SqlParameter("apellido", alumno.Apellido));
            command.Parameters.Add(new SqlParameter("correo", alumno.CorreoElectronico));
            command.Parameters.Add(new SqlParameter("fechanac", alumno.FechaNacimiento));
            command.Parameters.Add(new SqlParameter("contraseña", alumno.ContraseñaAlumno));
            command.Parameters.Add(new SqlParameter("horas", alumno.NumHoras));
            command.Parameters.Add(new SqlParameter("carrera", alumno.CarreraAlumno));
            alumnoGuardado = true;
        }
        conexionBaseDatos.CloseConnection();
    }

    return alumnoGuardado;
}

```

2. Aplicar las sangrías apropiadas a los segmentos de código que lo ameriten, por ejemplo, los ciclos, los selectivos y los condicionales.

```

namespace LogicaDeNegocio
{
    public class Expediente
    {
        public Documento DocumentoExpediente { get; set; }
        public Reporte ReporteExpediente { get; set; }
    }

    public class Documento
    {
        public String NombreDocumento { get; set; }
        public IdDocumento IdDocumento { get; set; }
        public String DescripcionDocumento { get; set; }
        public String FechaEntrega { get; set; }
    }

    public enum IdDocumento
    {
        CartaAsignacion = 1,
        CartaAceptacion = 2,
        FormatoRegistroYPlanActividades = 3,
        CartaLiberacion = 4,
        ReporteFinal = 5
    }
}

```

3. Respetar las nomenclaturas definidas para las variables, clases y métodos.

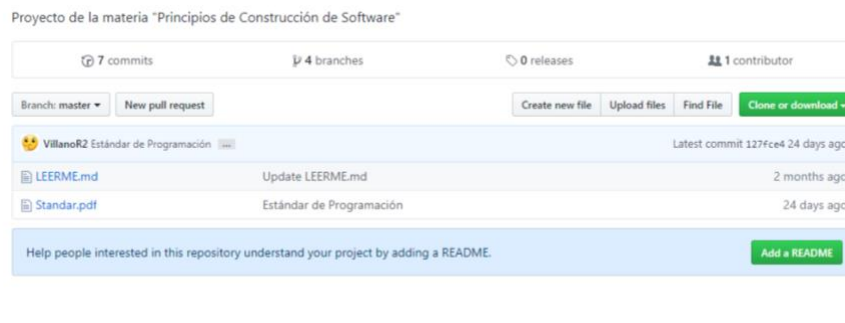
```

public class Alumno : Persona
{
    public String ContraseñaAlumno { get; set; }
    public String Matricula { get; set; }
    public int NumHoras { get; set; }
    public Carrera CarreraAlumno { get; set; }
}

public enum Carrera
{
    IngenieriaSoftware = 0,
    TecnologiasComputacionales = 1,
    RedesServiciosComputacionales = 2
}

```

4. Compartir el estándar con todo el equipo.



5. Usar solo los comentarios estrictamente necesarios.
No se han manejado comentarios hasta el momento.

6. Dejar espacios entre líneas de código y secciones diferentes.

```

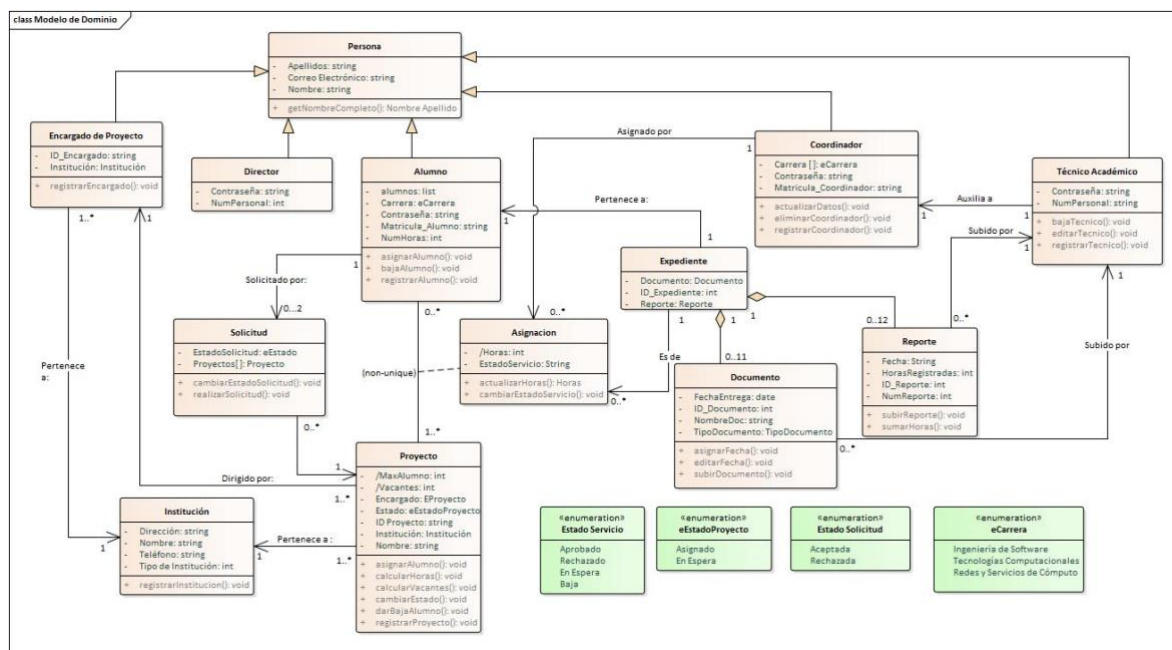
public bool asignarFechaADocumento(Documento documento)
{
    bool fechaGuardada = false;

    ConexionBaseDatos conexionBaseDatos = new ConexionBaseDatos();
    using (SqlConnection connection = conexionBaseDatos.GetConnection())
    {
        connection.Open();
        using (SqlCommand command = new SqlCommand("Update Documento set Fecha = @fecha where ID_Documento = @id;", connection))
        {
            command.Parameters.Add(new SqlParameter("fecha", documento.FechaEntrega));
            command.Parameters.Add(new SqlParameter("id", documento.IdDocumento));
            fechaGuardada = true;
        }
        conexionBaseDatos.CloseConnection();
    }

    return fechaGuardada;
}

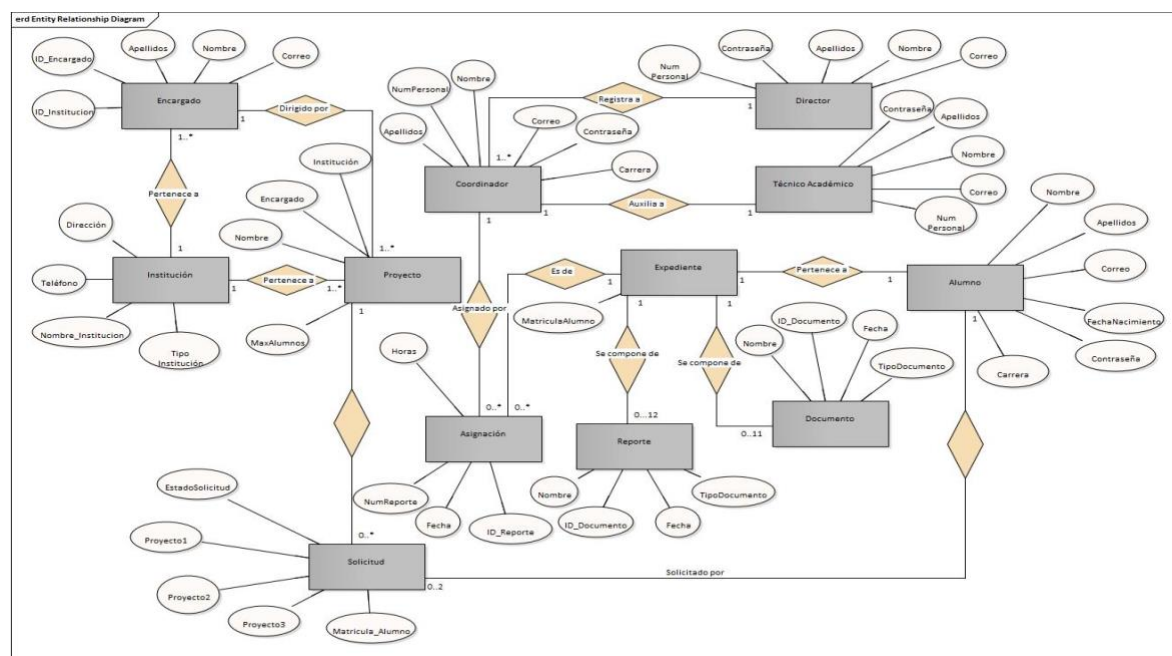
```

Modelo Dominio (Visto Bueno Diseño)



Modelo 1 Modelo De Dominio

Modelo Entidad Relación



Modelo 2 Modelo Entidad Relación

Prototipos

Este prototipo muestra la interfaz de inicio de sesión para el Sistema del Servicio Social FEI. La ventana principal tiene un título "Sistema Del Servicio Social FEI" y un logo de la Universidad Veracruzana. El formulario de inicio de sesión incluye campos para "Usuario" y "Contraseña", un botón "Iniciar Sesión" y un enlace "¿No tienes cuenta? [Regístrate como alumno](#)".

Sistema Del Servicio Social FEI

Universidad Veracruzana

Iniciar Sesión

Usuario:

Contraseña

Iniciar Sesión

¿No tienes cuenta? [Regístrate como alumno](#)

Pantalla 1 Pantalla de Inicio de sesión

Este prototipo muestra la interfaz principal para el Director del Sistema del servicio social. La ventana principal tiene un título "Sistema del servicio social: Director" y un usuario logueado. El formulario principal muestra una lista de coordinadores con columnas para Nombre, Carrera, Número de Personal y Correo Electrónico. Hay botones para "Registrar Coordinador" y "Eliminar Coordinador".

Sistema del servicio social: Director

Usuario:

Coordinadores


Nombre	Carrera	Número de Personal	Correo Electrónico
Giacomo Guizzoni Founder & CEO	40	Peldi	
Marco Botton Tuttofare	38		
Mariah MacLachlan Better Half	41	Patata	
Valerie Liberty Head Chef	;))	Val	

Registrar Coordinador


Eliminar Coordinador

Pantalla 2 Pantalla principal del director

Sistema del servicio social: Director

Usuario: 

Registrar Coordinadores

 Regresar

Nombre(s)

Apellidos


Grado de Estudios

Correo Electrónico

Carrera

Pantalla 3 Pantalla para registro de coordinador

Sistema del servicio social: Director

Usuario: 

Coordinadores

Nombre	Carrera	Número de Personal	Correo El
Giacomo Guilizzoni Founder & CEO	40	Peldi	
Marco Botton Tuttofare	38		
Mariah MacLachlan Better Half	41	Patata	
Valerie Liberty Head Chef	;))	Val	

Nombre

Número de Personal:

Pantalla 4 Pantalla principal director con ventana de "cerrar sesión".

Sistema del servicio social: Registro

Registro de Alumno

Nombre(s)

Apellidos

Matrícula

Licenciatura

Correo Electrónico

Contraseña

Confirmar contraseña


¿Tienes cuenta? [Iniciar Sesión](#)

Pantalla 5 Pantalla registro de alumno

Sistema del servicio social: Alumno

Nombre Usuario

Matrícula

Usuario: 

Su cuenta aun no es revisada por un coordinador. Porfavor intente más tarde.


Pantalla 6 Pantalla alumno pendiente de validación

Sistema del servicio social: Alumno

Nombre Usuario

Matricula

Usuario:



Mi Avance

Horas Reportadas al día

340/480

[Ver más...](#)

Proyecto

Organización

Facultad de Estadística e Informática

[Ver más...](#)


Próximos Eventos

APRIL 2019

S	M	T	W	T	F	S
	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				

Entrega primer reporte

/ /



[Ver Calendario Completo](#)

Pantalla 7 Pantalla principal alumno


Sistema del servicio social: Alumno

Nombre


Matricula

Carrera

Usuario:



Regresar



Documentos

Item One

Item Two

Item Three

Mes ▲	Numero Reporte ▼	Horas Reportadas ▲▼	Fecha en la que se subió

Número de horas realizadas al día de hoy:

Pantalla 8 Pantalla alumno documentos subidos

Sistema del servicio social: Alumno

Nombre Usuario

Matricula

Proyectos Disponibles

Proyecto 1

Proyecto 2

Proyecto 3

Proyecto 4

Opciones Escogidas:

Guardar

Proyecto 1

A paragraph of text with an [unassigned link](#).

A second row of text with a [web link](#)

Anterior

Siguiente

Usuario:

Próximos Eventos

APRIL 2019

S	M	T	W	T	F	S
	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				

No hay eventos que mostrar aún

Pantalla 9 Pantalla alumno solicitar servicio social

Sistema del servicio social: Alumno

Nombre Usuario

Matricula

Mi Avance

Horas Reportadas al día

340/480

Ver más...

Proyecto

Organización

Facultad de Estadística e Informática

Ver más...

Usuario:

Nombre

Matricula

Cerrar Sesión

Editar Perfil

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30


Entrega primer reporte

/ /

Ver Calendario Completo

Pantalla 10 Pantalla alumno principal

Sistema del servicio social: Coordinador

Nombre: Numero de personal: Carrera: Usuario: 

Periodo: Técnico Académico: [Más Opciones...](#)

Alumnos Inscritos


Nombre	Matricula	Correo Electrónico	Proyecto
Giacomo Guizzoni Founder & CEO	40	Peldi	
Marco Botton Tuttofare	38		
Mariah MacLachlan Better Half	41	Patata	
Valerie Liberty Head Chef	.)	Val	


Alumno Seleccionado

Nombre: Matricula: Avance:

Pantalla 11 Pantalla principal coordinador

Sistema del servicio social: Coordinador

Nombre: Numero de personal: Carrera: Usuario: 

 Regresar

Alumno

Nombre: Correo: Matricula: Proyecto:


Documentos


	Mes ▲	Numero Reporte ▼	Horas Reportadas ▲▼	Fecha en la que se subió
Item One				
Item Two				
Item Three				

Número de horas realizadas al día de hoy:

Pantalla 12 Pantalla Coordinador Alumno

Sistema del servicio social: Coordinador

Nombre Numero de personal Carrera Usuario: 

 Regresar

Técnico Académico

Nombre Correo Número de Personal

Registrar Técnico

Nombre(s)

Apellidos

Número de personal

Grado de estudios


Correo Electrónico

Contraseña

Confirmar contraseña

Pantalla 13 Pantalla Registro de Técnico

Sistema del servicio social: Coordinador

Nombre Numero de personal Carrera Usuario: 

Periodo: Técnico Académico: [Más Opciones..](#)

Alumnos Inscritos


Nombre	Matricula	Correo Electrónico	
Giacomo Guilizzoni Founder & CEO	40	Peldi	Ver más..
Marco Botton Tuttofare	38		Ver más..
Mariah MacLachlan Better Half	41	Patata	Ver más..
Valerie Liberty Head Chef	;))	Val	Ver más..

Pantalla 14 Pantalla coordinador con ventana de cerrar sesión

Sistema del servicio social: Coordinador

←

Regresar

Usuario: 

Guardar






























◀

APRIL 2019

▶

S	M	T	W	T	F	S
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	1	2	3	4
5	6	7	8	9	10	11

Descripción


B *I* U ~~S~~ style                             

Sistema del servicio social: Coordinador

Nombre

Numero de personal

Carrera

Usuario: 

Técnico Académico:

[Más Opciones...](#)

Validar Cuentas

Nombre	Matricula	Correo Electrónico	Validar
Giocomo Guizzoni Founder & CEO	40	Peldi	<input type="checkbox"/>
Marco Botton Tuttofare	38		<input type="checkbox"/>
Mariah Madachian Better Half	41	Patata	<input type="checkbox"/>
Valerie Liberty Head Chef	3	Val	<input type="checkbox"/>
			<input type="checkbox"/>

Cuentas Válidas

Nombre	Matricula	Correo Electrónico	
Giocomo Guizzoni Founder & CEO	40	Peldi	<input type="radio"/>
Marco Botton Tuttofare	38		<input type="radio"/>
Mariah Madachian Better Half	41	Patata	<input type="radio"/>
Valerie Liberty Head Chef	3	Val	<input type="radio"/>

Alumno Seleccionado

Nombre

Matricula

Proyectos

Proyecto 1

Proyecto 2


Proyecto 3


Proyectos Registrados

Nombre	Institución	Encargado	Número vacantes
Giocomo Guizzoni Founder & CEO	40	Peldi	
Marco Botton Tuttofare	38		
Mariah Madachian Better Half	41	Patata	
Valerie Liberty Head Chef	3	Val	

Pantalla 16 Principal Coordinador

Sistema del servicio social: Coordinador

Nombre Numero de personal Carrera Usuario: 

 Regresar

Registrar Institución

Nombre


Dirección


Teléfono

Tipo de Institución

Pantalla 17 Registrar institución

Sistema del servicio social: Coordinador

Nombre Numero de personal Carrera Usuario: 

 Regresar

Registrar Proyecto


Nombre


Institución

Número máximo de alumnos

Pantalla 18 Registrar proyecto

Sistema del servicio social: Coordinador

Nombre Numero de personal Carrera Usuario: 

 Regresar

Registrar Encargado

Nombre (s)

Apellido


Correo Electrónico


Institución

Proyecto

Pantalla 19 Registrar encargado

Sistema del servicio social: Coordinador

Nombre Numero de personal Carrera Usuario: 

 Regresar

Alumno Seleccionado

Nombre Matricula Proyectos


 Proyecto Asignado

Proyectos Disponibles

Nombre	Institución	Encargado	Número vacantes
Giacomo Guizzoni Founder & CEO	40	Peldi	<input type="radio"/>
Marco Botton Tuttofare	38		<input type="radio"/>
Mariah MacLachlan Better Half	41	Patata	<input type="radio"/>
Valerie Liberty Head Chef	0)	Val	<input type="radio"/>

Pantalla 20 Asignar Proyecto

Sistema del servicio social: Técnico Académico

Nombre Número de personal Carrera Usuario: 

Periodo:

Alumnos Inscritos

Nombre	Matricula	Correo Electrónico	
Giacomo Gullizzoni Founder & CEO	40	Peldi	<input type="radio"/>
Marco Botton Tuttofare	38		<input type="radio"/>
Mariah Madachian Better Half	41	Patata	<input type="radio"/>
Valerie Liberty Head Chef	3)	Val	<input type="radio"/>


Nombre Número de Personal:

Alumno Seleccionado

Nombre Matricula Avance

Pantalla 21 Coordinador principal con cerrar sesión

Sistema del servicio social: Técnico Académico

Nombre Número de personal Carrera Usuario: 

Periodo:

Alumnos Inscritos


Nombre	Matricula	Correo Electrónico	Proyecto
Giacomo Gullizzoni Founder & CEO	40	Peldi	
Marco Botton Tuttofare	38		
Mariah Madachian Better Half	41	Patata	
Valerie Liberty Head Chef	3)	Val	

Alumno Seleccionado

Nombre Matricula Avance

Pantalla 22 Principal Técnico

Sistema del servicio social: Técnico Académico


Nombre Numero de personal Carrera Usuario: 

Periodo:


Alumno Seleccionado

Nombre Matricula Avance

Tipo de Documento

 [nombre_documento.pdf](#)

Horas Reportadas

 [nombre_documento.pdf](#)

Pantalla 23 Subir documento

Explorador de Archivos

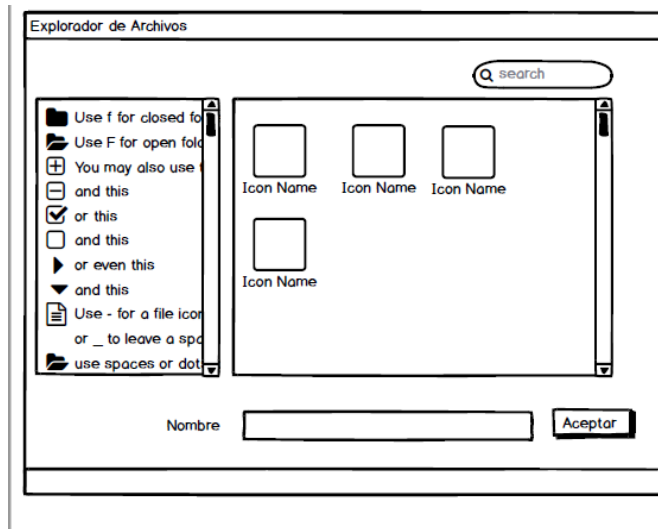
Use f for closed folder
Use F for open folder
You may also use
and this
or this
and this
or even this
and this
Use - for a file icon
or _ to leave a space
use spaces or dots

Icon Name Icon Name Icon Name

Icon Name

Nombre

Pantalla 24 Explorador de Archivos



Pantalla 25 Explorador de archivos

Propuesta de Capas

Nuestra propuesta para el sistema del Servicio Social se encuentra dividido en 3 capas:

Acceso a Datos:

- En esta capa se establece la conexión con la base de datos Servicio Social.

Lógica del Negocio

- En esta capa se encuentran codificadas todas las clases planteadas en el modelo de dominio así como sus relaciones.
- Se realizará la lectura y escritura en la base de datos.
- Se encuentran la carpeta acceso a datos: interfaces con la definición de los DAO, la definición de los mismos métodos; en la carpeta excepciones: las clases de manejo de excepciones personalizadas: la carpeta modelo con la definición de todas las clases y las clases enum por último en la carpeta Útil se encuentra una clase para validar usuarios y un clase convertidora de enums.

Servicio Social App

- En esta capa está todo aquello que será visible para el usuario.
- Pantallas en las que podrá ingresar datos y visualizar.

Documentacion de pruebas.

Nombre de Prueba	Fallida	Existosa	Observaciones
PruebaRegistrarAlumnoValido		✓	Se registra correctamente un alumno en la base de datos, se espera no obtener error alguno, regresa una confirmacion en "true".
PruebaRegistrarAlumnoValido2		✓	Se registra correctamente un alumno con datos diferentes a la prueba anterior en la base de datos, se espera no obtener error alguno, regresa una confirmacion en "true".
PruebaRegistrarAlumnoLlavePrimaria	X		Se intenta registrar un alumno con una matricula ya existente en la base de datos, se espera una excepcion a la llave duplicada, regresa una confirmacion en "false".
PruebaRegistrarAlumnoMatriculaNoValida	X		Se intenta registrar un alumno con una matricula que excede el limite de caracteres en la base de datos, se espera una excepcion a matricula no valida, regresa una confirmacion en "false".
PruebaRegistrarAlumnoLlavePrimariaVacía	X		Se intenta registrar un alumno sin ingresar ninguna matricula, se espera una excepcion a matricula vacia, regresa una confirmacion en "false".
PruebaRegistrarCoordinadorValido		✓	Se registra correctamente un coordinador en la base de datos, se espera no obtener error alguno, regresa una confirmacion en "true".
PruebaRegistrarCoordinadorMatriculaErronea	X		Se intenta registrar un coordinador con una matricula que excede el limite de caracteres en la base de datos, se espera una excepcion a matricula no valida, regresa una confirmacion en "false".
PruebaRegistrarCoordinadorMatriculaRepetida	X		Se intenta registrar un coordinador con una matricula ya existente en la base de datos, se espera una excepcion a la llave duplicada, regresa una confirmacion en "false".

PruebaRegistrarCoordinadorVacio	X		Se intenta registrar un coordinador sin ingresar ningun dato, se espera una excepcion a campos vacios, regresa una confirmacion en "false".
RegistrarEncargadoValido		✓	Se registra correctamente un encargado en la base de datos, se espera no obtener error alguno, regresa una confirmacion en "true".
RegistrarEncargadoRepetido	X		Se intenta registrar un encargado con una matricula ya existente en la base de datos, se espera una excepcion a la llave duplicada, regresa una confirmacion en "false".
RegistrarEncargadoCamposVacios	X		Se intenta registrar un encargado sin ingresar ningun dato, se espera una excepcion a campos vacios, regresa una confirmacion en "false".
PruebaRegistrarInstitucionValida		✓	Se registra correctamente una institucion en la base de datos, se espera no obtener error alguno, regresa una confirmacion en "true".
PruebaRegistrarInstitucionCamposVacios	X		Se intenta registrar una institución sin ingresar ningun dato, se espera una excepcion a campos vacios, res a una confirmacion en "false".
PruebaRegistrarInstitucionRepetida	X		Se intenta registrar una institucion ya existente en la base de datos, se espera una excepcion a la llave duplicada (El nombre de la institución funciona como llave unica), regresa una confirmacion en "false".
PruebaRegistrarTecnicoValido		✓	Se registra correctamente un tecnico académico en la base de datos, se espera no obtener error alguno, regresa una confirmacion en "true".
PruebaRegistrarTecnicoNoValido	X		Se intenta registrar una tecnico académico sin ingresar ningun dato, se espera una excepcion a campos vacios, regresa una confirmacion en "false".
PruebaRegistrarTecnicoLlaveRepetida	X		Se intenta registrar un tecnico académico con una matricula ya existente en la base de datos, se espera una excepcion a la llave duplicada, regresa una confirmacion en "false".
PruebaRegistrarTecnicoCoordinadorNoExiste	X		Se intenta registrar un tecnico académico con una matricula de un coordinador no existente en la base de datos, se espera una excepcion de coordinador no existente, regresa una confirmacion en "false".

--	--	--	--

Conclusiones

Arturo:

Gracias a el uso de pruebas unitarias se han detectado errores que al momento de compilar y ejecutar el software se pasaban por alto, añadiéndole un manejo de excepciones en diversos casos, se han solucionado algunos problemas que generaba al probar el código, también llegue a la conclusión que algunos fragmentos de código factibles al cambio y puede generar algunos problemas si no se maneja de manera unitaria los fragmentos de código que no dependen entre sí.

Regina:

Con el proyecto al día de hoy, se puede observar lo importante que es tener los modelos hechos en diseño, totalmente definidos ya que son la base para tomar muchas de las decisiones en el código y permite saber que métodos y atributos se encuentran en cada clase y como se encuentran relacionadas entre sí, lo que hace que la codificación sea una actividad casi de traducción de estos métodos. También noté que los prototipos son un gran apoyo ya que permite ver las ideas un poco más aterrizadas y poder tener un acercamiento al proyecto final, y llegar a esa meta realizando el código.

Referencias

El Estándar de Código está basado en el siguiente estándar de C#, escrito por José Luis Clemente Montes. <https://es.calameo.com/read/00443793525640a944420>

Uso de excepciones

<https://docs.microsoft.com/en-us/dotnet/standard/exceptions/best-practices-for-exceptions>

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/exceptions/>

<https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/exceptions/using-exceptions>

<https://albertcapdevila.net/control-excepciones-csharp/>

Uso de SQL en Visual Studio

<https://social.msdn.microsoft.com/Forums/es-ES/ad3cfe1e-7ec3-476f-9e0e-91a89f17cd02/hacer-una-consulta-sql-con-c-utilizando-un-data-reader-y-mostrarlo-por-consola?forum=vsexes>

<https://www.lawebdelprogramador.com/foros/C-sharp/1579952-Consultas-sql-en-C.html>

<https://solocodigo.com/30600/consulta-sql-desde-c/>