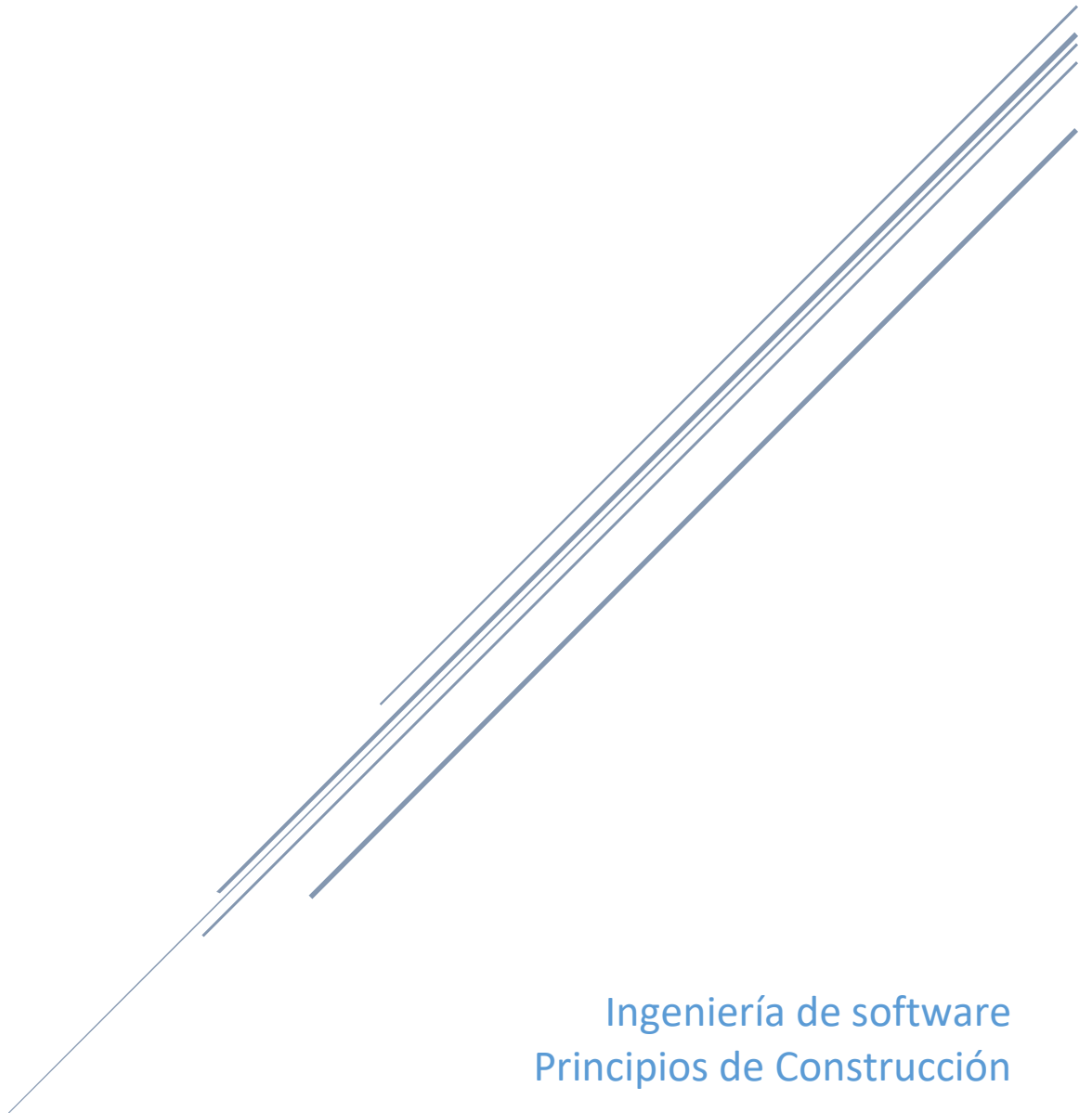


DOCUMENTACIÓN DEL PROYECTO FINAL

Sistema de Información de Servicio Social para la Facultad de
Estadística e Informática.



Ingeniería de software
Principios de Construcción

Kevin Yair Hernández Macías

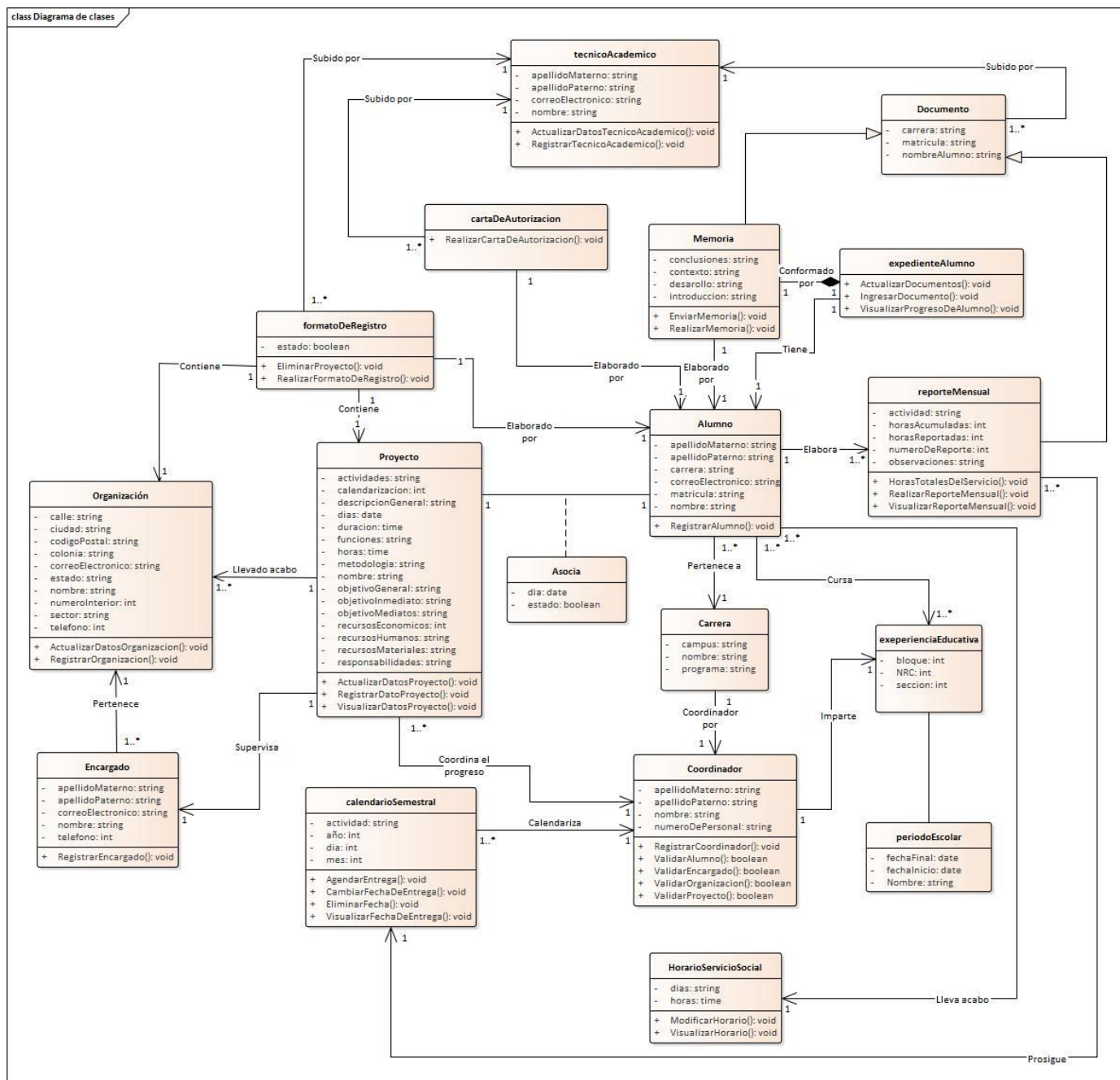
INTRODUCCIÓN.

La documentación del proyecto final de la experiencia educativa principios de construcción está centrada principalmente en la explicación de cómo está estructurado nuestro programa, así como también dar una visión en el enfoque de diseño de los módulos que fueron implementados dentro del sistema. El sistema consta de una serie de módulos elegidos y programados por los alumnos pertenecientes a la experiencia educativa principios de construcción, cada módulo fue realizado mediante la programación por capas la cual está constituida primordialmente por tres: capa lógica en la cual se establecen todas las reglas que deben cumplirse, capa de presentación la cual contiene las interfaces con las que el usuario interactuara y por último la capa de datos donde residen los datos y es la encargada de acceder a los mismos. El lenguaje de programación en cual fue desarrollado el sistema es C# junto con la elaboración de la base de datos utilizando SQL server.

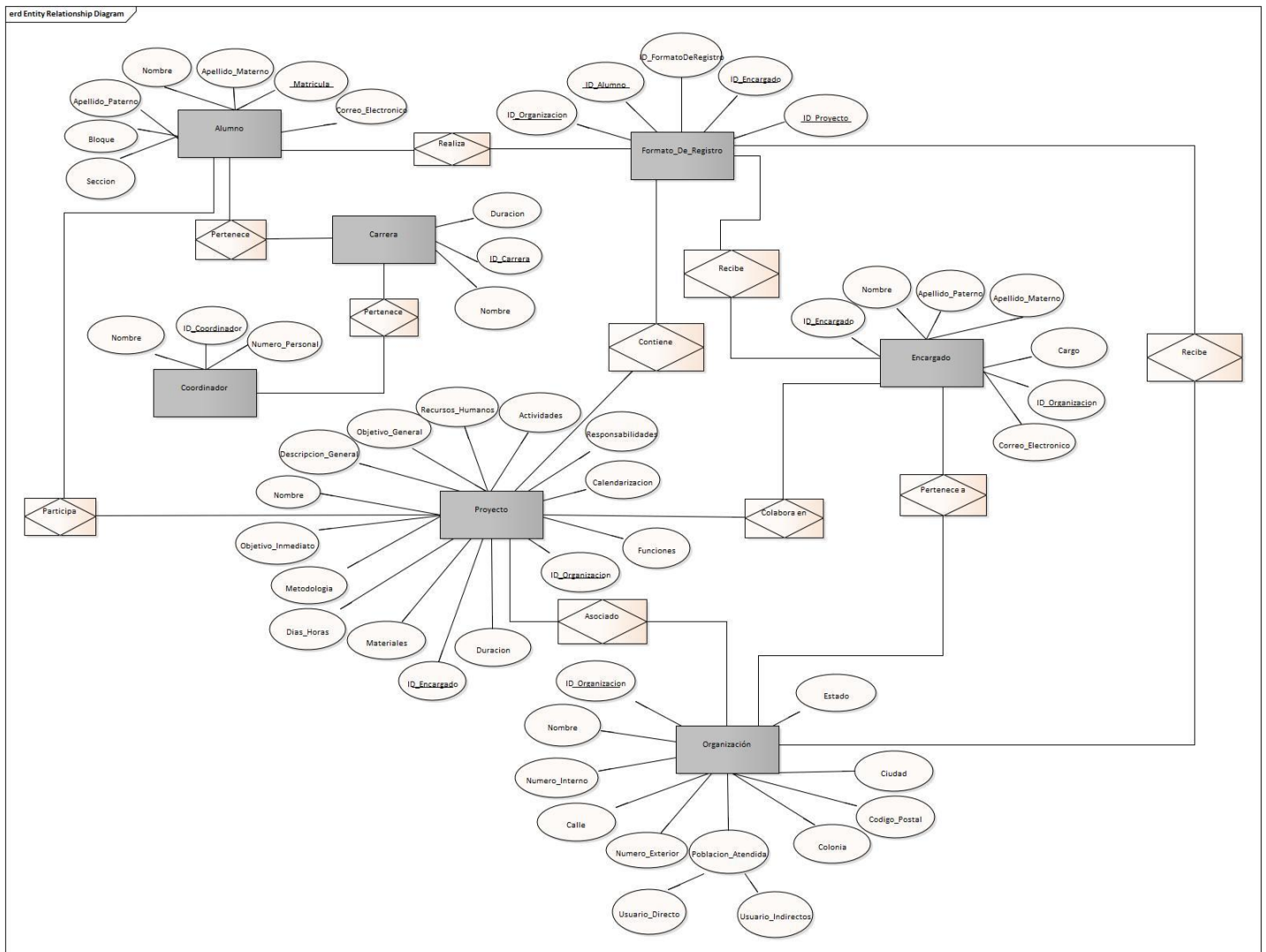
El sistema está constituido por los siguientes módulos: Enviar formato de registro, validar alumno, validar proyecto, validar encargado, validar organización, registrar proyecto, registrar encargado, registrar organización, registrar alumno; dentro del sistema se encuentran tres tipos de usuarios: el alumno, coordinador, encargado, cada usuario deberá registrarse dentro del sistema, así como también ser validados dentro del mismo.

Este documento tiene una estructura distribuida de la siguiente manera: Modelo de Dominio: se muestra la estructura general del sistema; Modelo Entidad Relación: se muestra la estructura la distribución de los datos dentro del sistema; Prototipos no funcionales: propuesta de interfaz de usuario para cada módulo; Estándar de codificación: muestra la manera que deberá estar estructurado el código fuente; Conclusión.

Diagrama de clases del modelo de dominio.



Modelo Entidad Relación.



Prototipos no funcionales.

Interfaz registrar alumno.

Al no contar con un usuario dentro del sistema el sistema desplegara la “IURegistrarAlumno” la cual contiene los campos mostrados en el prototipo.

Registro de alumno

Nombre:

Apellido paterno:

Apellido materno:

Matricula:

Correo Electronico:

Carrera:

Bloque:

Sección:

Interfaz registrar coordinador.

Al no contar con un usuario dentro del sistema el sistema desplegara la “IURegistrarCoordinador” la cual contiene los campos mostrados en el prototipo.

Registrar coordinador

Nombre:

Apellido Paterno:

Apellido Materno:

Numero Personal:

Interfaz registrar encargado.

Al no contar con un usuario dentro del sistema el sistema desplegara la “IURegistrarEncargado” la cual contiene los campos mostrados en el prototipo.

Prototipo de la interfaz "Registrar encargado". La ventana tiene un título "Registrar encargado" y botones de control de ventana. Contiene los siguientes campos de texto:

- Nombre(s):
- Apellido Paterno:
- Apellido Materno:
- Cargo:
- Organización a la que pertenece:
- Correo Electronico:

En la parte inferior hay dos botones: "Aceptar" y "Cancelar".

Interfaz registrar organización.

La siguiente interfaz muestra el formato de registro que desplegara el sistema para registrar una organización la cual contiene los campos de la interfaz “IURegistrarOrganizacion”.

Prototipo de la interfaz "Registrar organización". La ventana tiene un título "Registrar organización" y botones de control de ventana. Contiene los siguientes campos de texto:

- Nombre de la organización:
- Calle:
- Numero interno:
- Numero exterior:
- Colonia:
- Codigo postal:
- Ciudad:
- Estado:
- Telefono:
- Correo Electronico:
- Sector:

En la parte inferior hay dos botones: "Aceptar" y "Cancelar".

Interfaz registrar organización.

La siguiente interfaz muestra el formato de registro que desplegara el sistema para registrar un proyecto el cual contiene los campos de la interfaz “IURegistrarProyecto”.

The image shows a software window titled "Registrar proyecto" with standard Windows window controls (minimize, maximize, close). The form contains the following fields and labels:

- Nombre:** A text input field containing the placeholder text "Nombre".
- Descripción general:** A large text area for general description.
- Objetivo general:** A text input field for the general objective.
- Objetivo inmediato:** A text input field for the immediate objective.
- Metodología:** A text input field containing the placeholder text "Metodologia".
- Objetivo mediano:** A text input field for the medium-term objective.
- Recursos humanos:** A text input field for human resources.
- Recursos economicos:** A text input field for economic resources.
- Recursos materiales:** A text input field for material resources.
- Actividades y funciones:** A text input field containing the placeholder text "Actividades y funciones".
- Responsabilidades:** A text input field containing the placeholder text "Responsabilidades".
- Días y horas:** A text input field containing the placeholder text "Días y horas".
- Calendarización:** A text input field containing the placeholder text "Calendarización".

At the bottom right of the form are two buttons: "Enviar" and "Cancelar".

Estándar de codificación lenguaje C#.

Ajustes y estándares de nombres.

- Upper case: El primer carácter de todas las palabras en mayúsculas y los otros caracteres en minúsculas.

Ejemplo de código bueno	Ejemplo de código malo
HolaMundo.	holaMundo.

- Utilizar palabras entendibles y descriptivas. No usar abreviaciones.

Ejemplo de código bueno	Ejemplo de código malo
NumeroDeDíasEnMes.	Dias.

- No utilizar palabras reservadas.

Ejemplo de código bueno	Ejemplo de código malo
BaseDeDatos.	Base.

- Una excepción de la utilización de la declaración en una sola letra puede ser dentro de los ciclos como for, while, do while etc...

Ejemplo de código bueno	Ejemplo de código malo
<pre>public int DeterminarNumeroMaximo() { for(int i =0; i<numeroMaximo; i++) { //Sección de código } }</pre>	<pre>public int DeterminarNumeroMaximo() { Int i; for(i =0; i<numeroMaximo; i++) { //Sección de código } }</pre>

- No utilizar guiones bajo para la declaración de variables.

Ejemplo de código bueno	Ejemplo de código malo
IdentificadorDeEmpleado.	Identificador_De_Empleado.

- Usa el prefijo apropiado para cada elemento de la Interfaz Gráfica de manera que puedas identificarlos para el resto de las variables.

Ejemplo de código bueno	Ejemplo de código malo
IUMenuPrincipal.	Menu.

- Nombre de los archivos deben de coincidir con el nombre de la clase.

Ejemplo de código bueno	Ejemplo de código malo
<pre>ConexionBaseDeDatos class BaseDeDatos { //Codigo. }</pre>	<pre>ConexionBaseDeDatos class Datos { //Codigo. }</pre>

Sangría y espacios.

- Utilizar TAB para la sangría, no usar espacio.

Ejemplo de código bueno	Ejemplo de código malo
<pre>Public void Inicio() { //metodo }</pre>	<pre>Public void Inicio() { //metodo }</pre>

- Los comentarios deben estar al mismo nivel que el código (usar el mismo nivel de sangría).

Ejemplo de código bueno	Ejemplo de código malo
<pre>Public void Inicio() { //método //Comentario }</pre>	<pre>Public void Inicio() { //método //Comentario }</pre>

- Las llaves deben de estar declaradas una línea de código debajo de la declaración de algún método, ciclo, etc...

Ejemplo de código bueno	Ejemplo de código malo
<pre>String nombreDelUsuario() { }</pre>	<pre>String nombreDelUsuario(){ }</pre>

- Utiliza un enter después de la declaración de cada método.

Ejemplo de código bueno	Ejemplo de código malo
<pre>String nombreDelUsuario() { //Secciondecodigo }</pre>	<pre>String nombreDelUsuario() { //Secciondecodigo }</pre>

}	}
---	---

- Evita escribir métodos muy largos. Un método debe típicamente.

Ejemplo de código bueno	Ejemplo de código malo
String nombreDelUsuario() { //Secciondecodigo }	String nombreDelUsuario() { //Muchocodigo }

- Los nombres de los métodos deben decir lo que hacen.

Ejemplo de código bueno	Ejemplo de código malo
void GuardarNombreDelDocente()	Void Nombre()

- Un método debe de tener una sola tarea.

Ejemplo de código bueno	Ejemplo de código malo
void GuardarNombreDelDocente() { //Sumar }	void GuardarNombreDelDocente() { //Sumar //Restar //Multiplicar //Dividir }

- Evitar utilizar muchas variables, en la medida posible utiliza locales a menos que sean necesarias para otros métodos.
- Los mensajes de error deben de ayudar al usuario a resolver el problema.

Ejemplo de código bueno	Ejemplo de código malo
"Fallo en la actualización de la base de datos"	"Error"

- Evitar archivos muy grandes.

Ejemplo de código bueno	Ejemplo de código malo
Proyecto=10mb	Proyecto=300mb

- Evita métodos y propiedades públicas, a menos que ellas realmente necesiten ser accedidas desde afuera de la clase.

Ejemplo de código bueno	Ejemplo de código malo
private String NombreAlumno()	public String NombreAlumno()

- No utilizar muchos parámetros en un método.

Ejemplo de código bueno	Ejemplo de código malo
<pre>private String NombreAlumno(String Nombre, String ApellidoPaterno, String ApellidoMaterno)</pre>	<pre>public String NombreAlumno(String Nombre, String ApellidoPaterno, String ApellidoMaterno, int Edad, double Peso,)</pre>

- Los métodos solo deben de realizar una tarea a la vez (la tarea que tiene por nombre).

Ejemplo de código bueno	Ejemplo de código malo
<pre>ValidarCampos() { //Codigo } RegistrarUsuario() { //Codigo }</pre>	<pre>ValidarYRegistrarUsuario() { //Codigo }</pre>

- Organizar lógicamente los archivos dentro de carpetas apropiadas.

Ejemplo de código bueno	Ejemplo de código malo
<pre>LogicaDeNegocios Interfa BaseDeDatos PruebasUnitarias</pre>	<pre>Colocación de archivos no deseados en apartados no debidos: LogicaDeNegocios ConexionBaseDeDatos.cs</pre>

- Si estas abriendo conexiones a una base de datos, sockets, archivos etc, siempre cierra dichas conexiones.

Ejemplo de código bueno	Ejemplo de código malo
<pre>command.Parameters.Add(new SqlParameter("IDCoordinador", idCoordinador)); SqlDataReader reader = command.ExecuteReader(); while (reader.Read()) { coordinador.Nombre = reader["Nombre"].ToString();</pre>	<pre>command.Parameters.Add(new SqlParameter("IDCoordinador", idCoordinador)); SqlDataReader reader = command.ExecuteReader(); while (reader.Read()) { coordinador.Nombre = reader["Nombre"].ToString();</pre>

<pre> coordinador.ApellidoPaterno = reader["ApellidoPaterno"].ToString(); coordinador.ApellidoMaterno = reader["ApellidoMaterno"].ToString(); coordinador.NumeroPersonal = (int)reader["NumeroPersonal"]; } } dbManager.CerrarConexion(); </pre>	<pre> coordinador.ApellidoPaterno = reader["ApellidoPaterno"].ToString(); coordinador.ApellidoMaterno = reader["ApellidoMaterno"].ToString(); coordinador.NumeroPersonal = (int)reader["NumeroPersonal"]; } } </pre>
---	--

- Declarar variables lo más cerca posible de donde son usadas. Usa una declaración de variable por línea.

Ejemplo de código bueno	Ejemplo de código malo
<pre> ContarCadena() { int contador; if(//codigo) { contador++; } } </pre>	<pre> ContarCadena() { int contador; if(//codigo) { if(//Codigo) { if(//Codigo) { contador++; } } } } </pre>

Ordenamiento de paquetes y archivos.

- Los paquetes deben ser ordenados por orden alfabético.

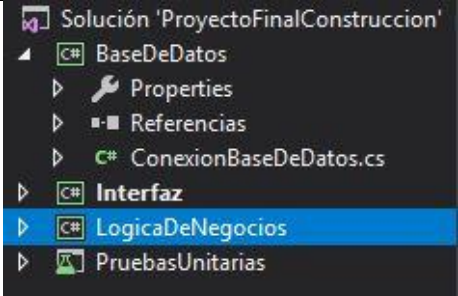
Ejemplo de código bueno	Ejemplo de código malo
<ul style="list-style-type: none"> ○ BaseDeDatos. ○ Interfaz. ○ LogicaDeNegocios. 	<ul style="list-style-type: none"> ○ PruebasUnitarias. ○ Interfaz. ○ BaseDeDatos.

○ PruebasUnitarias.	○ LogicaDeNegocios.
---------------------	---------------------

- Cada paquete debe contener nombres descriptivos.

Ejemplo de código bueno	Ejemplo de código malo
<ul style="list-style-type: none"> ○ BaseDeDatos. ○ Interfaz. ○ LogicaDeNegocios. ○ PruebasUnitarias. 	<ul style="list-style-type: none"> ○ Pruebas. ○ Interfaz. ○ Conexion. ○ Logica.

- Los paquetes deben de tener solo las clases utilizadas en él.

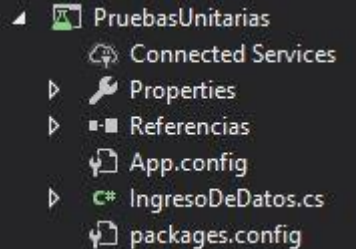
Ejemplo de código bueno	Ejemplo de código malo
	

- Particionar las actividades en distintas clases (No incluir demasiado código por clase).

Ejemplo de código bueno	Ejemplo de código malo
<pre> namespace LogicaDeNegocios { public class ValidacionDeInterfaz { public static Boolean ValidarTextbox(UIElementCollection ElementoColeccion) { Boolean bandera = false; foreach (UIElement ElementoInterfaz in ElementoColeccion) { if (ElementoInterfaz is TextBox) { TextBox Textbox = (TextBox)ElementoInterfaz; if (Textbox.Text == "") { bandera = false; break; } else { bandera = true; } } } return bandera; } } } </pre>	

Pruebas Unitarias.

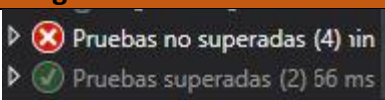
- Todas las pruebas estarán en un paquete específico.

Ejemplo de código bueno	Ejemplo de código malo
	

- Cada prueba solo probará una sección o método de una clase no la clase completa.

Ejemplo de código bueno	Ejemplo de código malo
<pre>[TestMethod] public void RegistrarEncargado() { Encargado NuevoEncargado = new Encargado(); NuevoEncargado.Nombre = "Marco"; NuevoEncargado.ApellidoPaterno = "Hernandez"; NuevoEncargado.ApellidoMaterno = "Luna"; NuevoEncargado.CorreoElectronico = "marco@lania.mx"; NuevoEncargado.IDEncargado = "590038993"; NuevoEncargado.Cargo = "Administrador del control escolar"; int resultado = EncargadoDAO.RegistrarDatosEncargado(NuevoEncargado); Assert.AreEqual(resultado, 1); }</pre>	

- Todas las pruebas deben pasar como aprobadas o en su defecto si su funcionalidad es comprobar que el sistema no deja que esa prueba se ejecute como aprobada.

Ejemplo de código bueno	Ejemplo de código malo
	

Interfaz de usuario.

- Cada interfaz de usuario deberá presentar al inicio de su pronunciación el sufijo "IU" para identificar que es una interfaz de usuario.

Ejemplo de código bueno	Ejemplo de código malo
IURegistrarUsuario. IUMostrarFormato.	RegistrarUsuario. MostrarFormato.

- Para el uso de algún TextBlock se usará el sufijo "TextField" seguido del nombre que se desea colocar, esto con el fin de diferenciarlos de los TextBox.

Ejemplo de código bueno	Ejemplo de código malo
TextFieldNombre.	Nombre.

TextFieldApellido.	Apellido.
TextFieldNumeroTelefonico	NumeroTelefonico.

- Para el uso de algún TextBox se usará el sufijo “TextBox” seguido del nombre que se desea colocar, esto con el fin de diferenciarlos de los TextBlock.

Ejemplo de código bueno	Ejemplo de código malo
TextBoxNombre.	Nombre.
TextBoxApellido.	Apellido.
TextBoxNumeroTelefonico	NumeroTelefonico.

- Para el uso de algún Button se usará el sufijo “Button” seguido del nombre que se desea colocar.

Ejemplo de código bueno	Ejemplo de código malo
ButtonAgregar.	Agregar.
ButtonAceptar.	Aceptar.
ButtonRegresar.	Regresar.

- Al utilizar un DataGrid se utilizará el sufijo “DataGrid” seguido del nombre que se desea colocar.

Ejemplo de código bueno	Ejemplo de código malo
DataGridConsultaAlumno.	ConsultaAlumno.
DataGridConsultaEncargado.	ConsultaEncargado.
DataGridConsultaProyecto.	ConsultaProyecto.

Casos de Pruebas.

Nombre del Tester:	Kevin Yair Hernández Macías		
Fecha:	20/06/2019		
Entradas de prueba.	Salida esperada.	Salida obtenida.	Resultado (Paso-No paso).
Entrada de registros.			
NuevoEncargado.Nombre = "Marco"; NuevoEncargado.ApellidoPaterno = "Hernandez"; NuevoEncargado.ApellidoMaterno = "Luna"; NuevoEncargado.CorreoElectronico = "marco@lania.mx"; NuevoEncargado.IDEncargado = "S90038993"; NuevoEncargado.Cargo = "Administrador del control escolar";	Prueba superada.	Prueba superada.	Paso.
NuevoAlumno.Nombre = "Marcela"; NuevoAlumno.ApellidoPaterno = "Días"; NuevoAlumno.ApellidoMaterno = "Gonzales"; NuevoAlumno.CorreoElectronico = "Marcela_Gonzales@gmail.com"; NuevoAlumno.Matricula = "S16011731"; NuevoAlumno.Seccion = "2"; NuevoAlumno.Bloque = "5";	Prueba superada.	Prueba superada.	Paso.
NuevoCoordinador.Nombre = "Juan Carlos"; NuevoCoordinador.ApellidoPaterno = "Perez"; NuevoCoordinador.ApellidoMaterno = "Arriaga"; NuevoCoordinador.IDCoordinador = "S90050281"; NuevoCoordinador.NumeroPersonal = 90050281;	Prueba superada.	Prueba superada.	Paso.
NuevaOrganizacion.Nombre = "LANIA"; NuevaOrganizacion.IDOrganizacion = "400178349"; NuevaOrganizacion.NumeroExterno = 20;	Prueba superada.	Prueba superada.	Paso.

<pre> NuevaOrganizacion.NumeroInterno = 0; NuevaOrganizacion.PoblacionAtendida = 100; NuevaOrganizacion.Telefono = 8416100; NuevaOrganizacion.UsuarioDirecto = "Cora"; NuevaOrganizacion.UsuarioIndirecto = "Cristina"; NuevaOrganizacion.Calle = "Rebsamen"; NuevaOrganizacion.Estado = "Veracruz"; NuevaOrganizacion.Ciudad = "Xalapa"; NuevaOrganizacion.CodigoPostal = "91190"; NuevaOrganizacion.CorreoElectronico = "Lania_Dudas@lania.mx"; NuevaOrganizacion.IDOrganizacion = "10035501"; NuevaOrganizacion.Colonia = "Centro"; </pre>			
<pre> NuevoProyecto.Nombre = "Sistema De Vialidad Publica"; NuevoProyecto.IDProyecto = "100904523"; NuevoProyecto.ObjetivoGeneral = "Crear un sistema que ayude a los estudiantes a escoger un transporte publico que los lleve a su destino"; NuevoProyecto.ObjetivoInmediato = "Realizacion de la parte de diseño con respecto al sistema"; NuevoProyecto.RecursoHumano = "Estudiantes de la licenciatura en ingenieria de software"; NuevoProyecto.RecursoMaterial = "Equipo de computo de la facultad"; NuevoProyecto.Responsabilidad = "Realizacion de los modulos indicados por el encargado de la organizacion"; </pre>	Prueba superada.	Prueba superada.	Paso.

<pre> NuevoProyecto.Metodologia = "Metodologia de prototipo"; NuevoProyecto.Duracion = 8; NuevoProyecto.Calendarizacion = "2019-06-20"; NuevoProyecto.DescripcionGeneral = "Sistema que ayude al desplazamiento de los alumnos de la Universidad Veracruzana"; NuevoProyecto.Actividad = "Las propocionadas por el Coordinador"; NuevoProyecto.Funcion = "Diseño e implementacion"; </pre>			
<pre> NuevoEncargado.Nombre = "M"; NuevoEncargado.ApellidoPaterno = "H"; NuevoEncargado.ApellidoMaterno = "L"; NuevoEncargado.CorreoElectronico = "m"; NuevoEncargado.IDEncargado = "S"; NuevoEncargado.Cargo = "A"; </pre>	Prueba superada.	Prueba superada.	Paso.
<pre> NuevoAlumno.Nombre = "M"; NuevoAlumno.ApellidoPaterno = "D"; NuevoAlumno.ApellidoMaterno = "G"; NuevoAlumno.CorreoElectronico = "b"; NuevoAlumno.Matricula = "S"; NuevoAlumno.Seccion = "2"; NuevoAlumno.Bloque = "5"; </pre>	Prueba superada.	Prueba superada.	Paso.
<pre> NuevoCoordinador.Nombre = "A"; NuevoCoordinador.ApellidoPaterno = "P"; NuevoCoordinador.ApellidoMaterno = "A"; NuevoCoordinador.IDCoordinador = "S9"; NuevoCoordinador.NumeroPersonal = 9; </pre>	Prueba superada.	Prueba superada.	Paso.
<pre> NuevaOrganizacion.Nombre = "L"; NuevaOrganizacion.IDOrganizacion = "1"; </pre>	Prueba superada.	Prueba superada.	Paso.

NuevaOrganizacion.NumeroExterno = 1; NuevaOrganizacion.NumeroInterno = 0; NuevaOrganizacion.PoblacionAtendida = 1; NuevaOrganizacion.Telefono = 1; NuevaOrganizacion.UsuarioDirecto = "C"; NuevaOrganizacion.UsuarioIndirecto = "C"; NuevaOrganizacion.Calle = "R"; NuevaOrganizacion.Estado = "V"; NuevaOrganizacion.Ciudad = "X"; NuevaOrganizacion.CodigoPostal = "9"; NuevaOrganizacion.CorreoElectronico = "L"; NuevaOrganizacion.IDOrganizacion = "1"; NuevaOrganizacion.Colonia = "C";			
"NuevoProyecto.Nombre = "S"; NuevoProyecto.IDProyecto = "1"; NuevoProyecto.ObjetivoGeneral = "C"; NuevoProyecto.ObjetivoInmediato = "R"; NuevoProyecto.RecursoHumano = "E"; NuevoProyecto.RecursoMaterial = "E"; NuevoProyecto.Responsabilidad = "R"; NuevoProyecto.Metodologia = "M"; NuevoProyecto.Duracion = 8; NuevoProyecto.Calendarizacion = "2019-06-20"; NuevoProyecto.DescripcionGeneral = "S"; NuevoProyecto.Actividad = "L"; NuevoProyecto.Funcion = "D";	Prueba superada.	Prueba superada.	Paso.
NuevoEncargado.Nombre = "Eduardo";	Prueba no superada.	Prueba no superada.	Paso.

NuevoEncargado.ApellidoPaterno = " Lopez"; NuevoEncargado.ApellidoMaterno = " Sanchez"; NuevoEncargado.CorreoElectronico = " eduardo_administrador@hotmail.com"; NuevoEncargado.IDEncargado = " S900389934"; NuevoEncargado.Cargo = " Administrador de contaduria";	"Error en la base de datos: Longitud del dato sobre pasado"	"Error en la base de datos: Longitud del dato sobre pasado"	
NuevoAlumno.Nombre = " Carmen"; NuevoAlumno.ApellidoPaterno = " Gomez"; NuevoAlumno.ApellidoMaterno = " Palacio"; NuevoAlumno.CorreoElectronico = " Gomez1998@gmail.com"; NuevoAlumno.Matricula = " S160117312"; NuevoAlumno.Seccion = " 2"; NuevoAlumno.Bloque = " 5";	Prueba no superada. "Error en la base de datos: Longitud del dato sobre pasado"	Prueba no superada. "Error en la base de datos: Longitud del dato sobre pasado"	Paso.
NuevoCoordinador.Nombre = " Jorge Ocatvio"; NuevoCoordinador.ApellidoPaterno = " Ocharan"; NuevoCoordinador.ApellidoMaterno = " Hernández"; NuevoCoordinador.IDCoordinador = " S900502813"; NuevoCoordinador.NumeroPersonal = 90050281;	Prueba no superada. "Error en la base de datos: Longitud del dato sobre pasado"	Prueba no superada. "Error en la base de datos: Longitud del dato sobre pasado"	Paso.
NuevoProyecto.Nombre = " Sistema De Vialidad Publica"; NuevoProyecto.IDProyecto = " 1009045235"; NuevoProyecto.ObjetivoGeneral = " Crear un sistema que ayude a los estudiantes a escoger un transporte publico que los lleve a su destino"; NuevoProyecto.ObjetivoInmediato = " Realizacion de la parte de diseño con respecto al sistema"; NuevoProyecto.RecursoHumano =	Prueba no superada. "Error en la base de datos: Longitud del dato sobre pasado"	Prueba no superada. "Error en la base de datos: Longitud del dato sobre pasado"	Paso.

<p>"Estudiantes de la licenciatura en ingeniería de software";</p> <p>NuevoProyecto.RecursoMaterial = "Equipo de computo de la facultad";</p> <p>NuevoProyecto.Responsabilidad = "Realización de los módulos indicados por el encargado de la organización";</p> <p>NuevoProyecto.Metodologia = "Metodología de prototipo";</p> <p>NuevoProyecto.Duracion = 8;</p> <p>NuevoProyecto.Calendarizacion = "2019-06-20";</p> <p>NuevoProyecto.DescripcionGeneral = "Sistema que ayude al desplazamiento de los alumnos de la Universidad Veracruzana";</p> <p>NuevoProyecto.Actividad = "Las propocionadas por el Coordinador";</p> <p>NuevoProyecto.Funcion = "Diseño e implementacion";</p>			
<p>NuevaOrganizacion.Nombre = "LANIA";</p> <p>NuevaOrganizacion.IDOrganizacion = "4001783492";</p> <p>NuevaOrganizacion.NumeroExterno = 20;</p> <p>NuevaOrganizacion.NumeroInterno = 0;</p> <p>NuevaOrganizacion.PoblacionAtendida = 100;</p> <p>NuevaOrganizacion.Telefono = 8416100;</p> <p>NuevaOrganizacion.UsuarioDirecto = "Cora";</p> <p>NuevaOrganizacion.UsuarioIndirecto = "Cristina";</p> <p>NuevaOrganizacion.Calle = "Rebsamen";</p> <p>NuevaOrganizacion.Estado = "Veracruz";</p> <p>NuevaOrganizacion.Ciudad = "Xalapa";</p> <p>NuevaOrganizacion.CodigoPostal = "91190";</p>	<p>Prueba no superada.</p> <p>"Error en la base de datos: Longitud del dato sobre pasado</p>	<p>Prueba no superada.</p> <p>"Error en la base de datos: Longitud del dato sobre pasado</p>	<p>Paso.</p>

<pre>NuevaOrganizacion.CorreoElectronico = "Lania_Dudas@lania.mx"; NuevaOrganizacion.IDOrganizacion = "10035501"; NuevaOrganizacion.Colonia = "Centro";</pre>			
---	--	--	--

Conclusión.

En el transcurso del curso hemos tocado puntos importantes en cuanto a ciertos aspectos referentes a una buena codificación. Sin embargo, el solo tener conocimiento de ello no nos hace ser buenos programadores ya que el reflejo de lo aprendido durante el curso se ve plasmado en la ejecución de dichos pasos para la buena codificación. Para ello la Experiencia Educativa Principios de Construcción nos impulsa a la realización de un proyecto integral donde aplicamos los aspectos aprendidos durante el curso, con el fin de poner en práctica los conocimientos adquiridos durante el curso.

Durante la elaboración de dicho proyecto se aplicaron los conocimientos adquiridos durante el curso, con el entendimiento de realizar código de calidad, en resumidas cuentas, desde el punto de vista personal aprendí a nombrar las variables de manera adecuada, a estructural el código en capas séase (Lógica de negocios, Interfaz de usuario, Base de datos) lo cual divide el sistema para protección de nuestro código, manejo eficaz de excepciones, pruebas unitaria etc... Puedo llegar a la conclusión de que los conceptos manejados durante el curso fueron puestos a prueba y pasados con éxito con la elaboración del proyecto integrador.