UTRM

Hectorb.peraza@gmail.com

Asignatura: Calidad En el desarrollo de software

Hector Benjamín Pech Peraza

universidad tecnologia de la riviera maya

TSU en Tecnologías de la Información TI 51

Contenido

[Planeación. 2](#_Toc474941665)

[Objetivos 2](#_Toc474941666)

[Alcance 2](#_Toc474941667)

[Requerimientos 2](#_Toc474941668)

[Diseño 2](#_Toc474941669)

[Estructuras de control 4](#_Toc474941670)

[If 4](#_Toc474941671)

[Switch 4](#_Toc474941672)

[Go to 4](#_Toc474941673)

[Estructuras loops / Iterativas 4](#_Toc474941674)

[Do 4](#_Toc474941675)

[Do while 4](#_Toc474941676)

[While 4](#_Toc474941677)

[For 4](#_Toc474941678)

[Foreach 4](#_Toc474941679)

[Clases 4](#_Toc474941680)

[Método / función 5](#_Toc474941681)

[Variables / Propiedades 5](#_Toc474941682)

[Constantes 5](#_Toc474941683)

[Comentarios 5](#_Toc474941684)

[Codificación 5](#_Toc474941685)

# Planeación.

## Objetivos

Crear una aplicación que permita la gestión de usuarios administrándolos por categorías, el usuario podrá realizar funciones como lo son:

* Agregar
* Eliminar
* Modificar

Así mismo el sistema debe determinar si los datos ingresados son correctos.

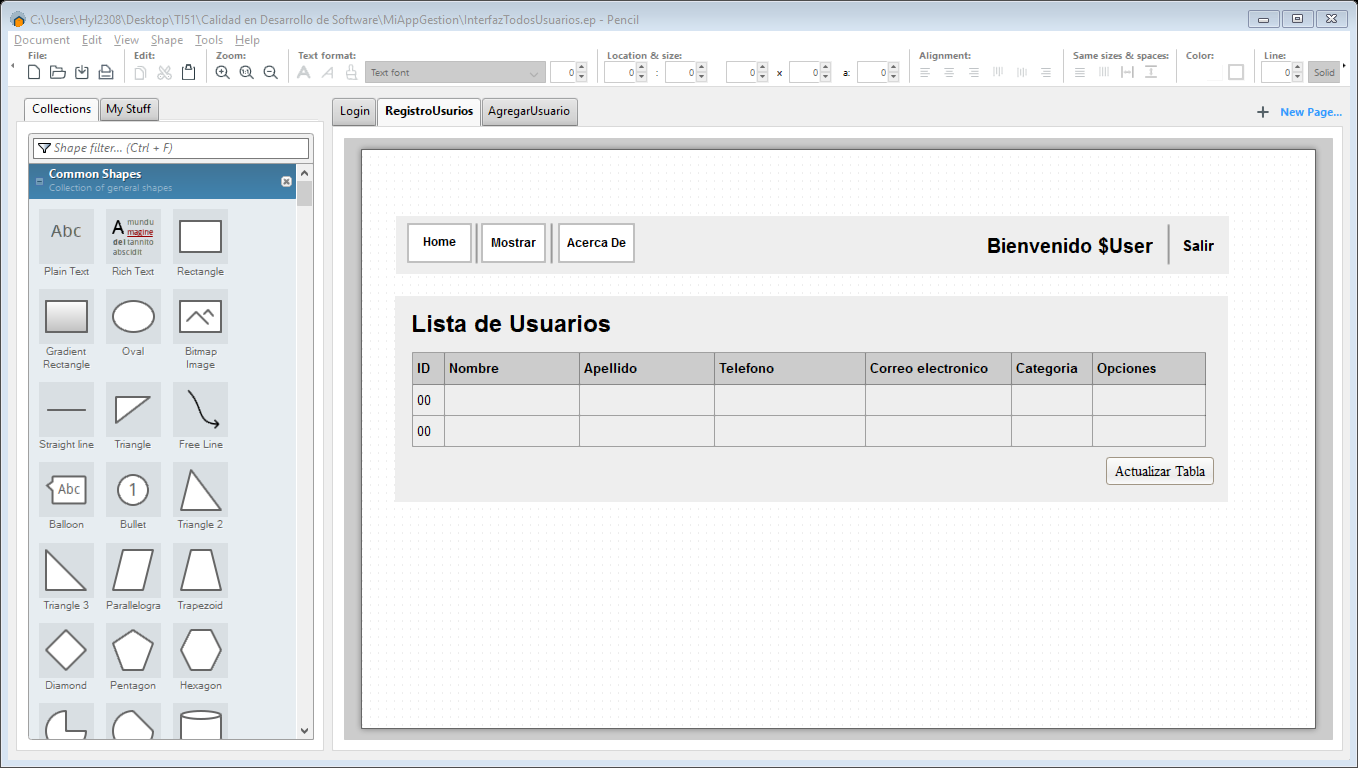
## Alcance

Esta aplicación permitirá visualizar una lista de usuarios agregados a la base de datos así mismo permitirá agregar, eliminar y modificar los usuarios ya registrados previamente.

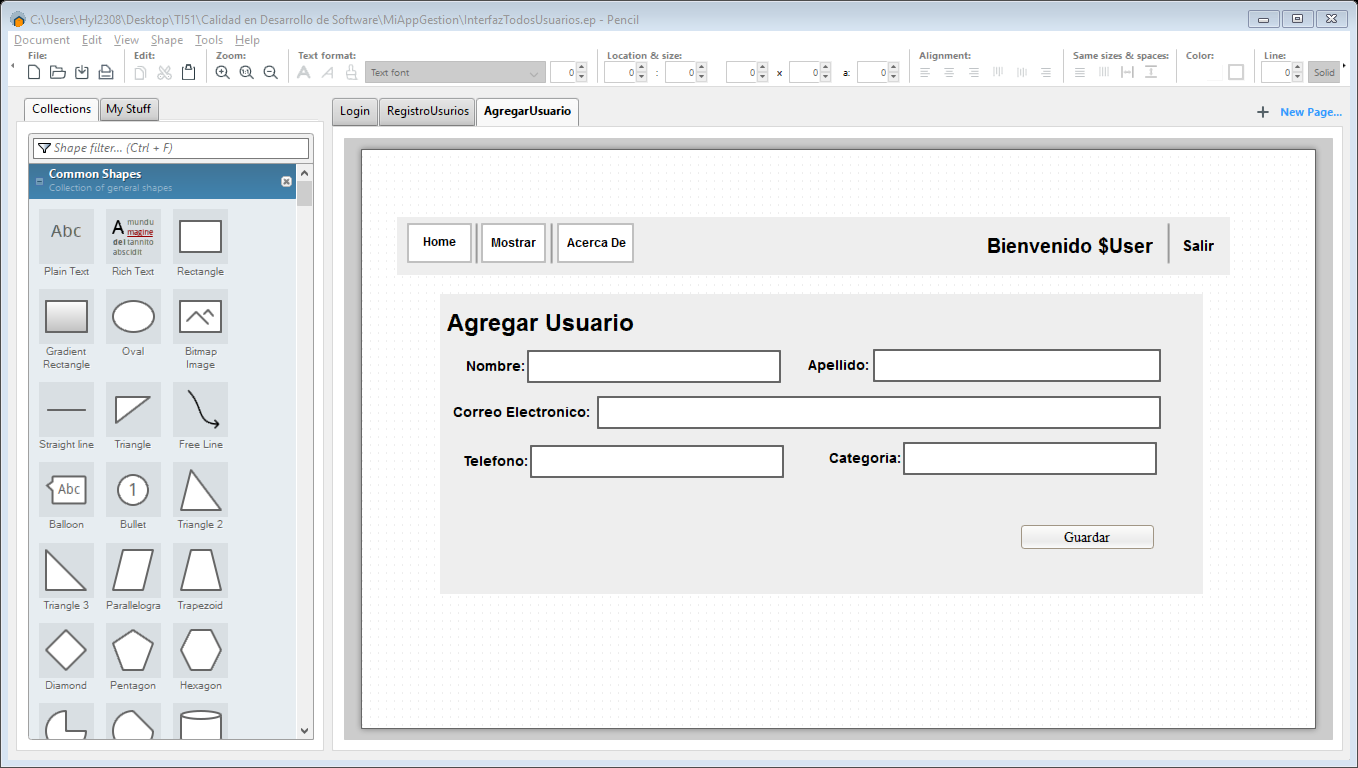
## Requerimientos

* Determinar si el correo electrónico ingresado valido
* Determinar si los valore de los campos son validos
* No permitir ingresar nombre y apellidos nulos
* No permitir que el campo de categorías este vacío
* Los correos electrónicos no deben repetirse

# Diseño



2 Interfaz de visualización de usuarios



3 Interfaz Agregar Usuario

**Convención de nombres**

Los archivos con la extensión .php se nombraran con la primera letra en mayúscula y en caso de tener mas palabras se pondrá en mayúscula la primera letra.

**Diseño de archivos**

Las carpetas utilizadas tendrán una sola palabra y la primera letra iniciara en mayúscula, se creara el nombre de manera singular evitando el uso de palabras plurales.

Los archivos que contengan código contendrán la extensión .php

**Diseño de clases**

Las clases serán colocadas en un archivo .php aparte, donde sólo se colocará el código de la clase. El nombre del archivo será el mismo del de la clase y siempre empezará en mayúscula. En lo posible, procurar que los nombres de clase tengan una sola palabra.

Las clases siguen las mismas reglas de las funciones, por tanto, debe colocarse un comentario antes de la declaración de la clase explicando su utilidad.

Las clases serán declaradas de tal forma que la primera letra de la palabra sea en mayúscula y si es nombrada con otra palabra agregar una separación con un guion bajo (\_) y las llaves serán colocadas un salto de línea después del nombre de la clase.

**Diseño de funciones**

El nombre de las funciones se declarara sin espacios en caso de tener más de una, se colocara la primera letra de cada palabra en mayúsculas.

Las funciones no deben devolver más de 1 solo valor

**Diseño de variables**

Las variables serán declaradas con la primera letra en mayúsculas y el resto en minúsculas

Si la variable contiene 2 palabras la segunda tendrá la primera letra en mayúscula

**Diseño de las tablas (DB)**

El nombre de la tabla de la base de datos se escribirá de forma plural y los campos se escribirán de forma singular, si es necesario escribir el nombre usando 2 palabras se separara a través de un guion bajo (\_).

**Comentarios**

Es importante que todos los archivos .php inicien con una cabecera específica que indique información de la versión, autor de los últimos cambios, etc.

## Estructuras de control

### If

If($Numero<0)

{

}

### Switch

### Go to

## Estructuras loops / Iterativas

### Do

### Do while

do

{

}while(($Valor<10);

### While

while($Valor<10)

{

}

### For

for($I = 0 ; $I ==10 ; $I++)

{

}

### Foreach

foreach($Data as $Valores)

{

}

## Clases

## Método / función

public function CalcularEdad()

{

Operaciones

}

## Variables / Propiedades

$Variable;

## Constantes

define(‘PI’,3.1416);

## Comentarios

/\*\*

Sistema de Gestión de Usuarios

Autor: Hector Pech

Versión: 1.1 Última modificación: 00/00/0000

\*/

# Codificación

El siguiente código desempeña la función de obtener los parámetros de la barra de búsqueda del navegador para re direccionar al usuario a una página determinada. El controlador **Categorías** desempeña funciones similares.

include\_once("Modelo/Usuarios.php");

class UsuariosControlador

{

private $Modelo;

function \_\_construct()

{

$this->Modelo= new Usuario;

}

public function index(){

$Data=$this->Modelo->ObtenerDatos();

include\_once("Vistas/Default/header.php");

include\_once("Vistas/Usuarios/index.php");

include\_once("Vistas/Default/footer.php");

}

public function NuevoUsuario(){

$Categorias=$this->Modelo->ObtenerCategorias();

include\_once("Vistas/Default/header.php");

include\_once("Vistas/Usuarios/NuevoUsuario.php");

include\_once("Vistas/Default/footer.php");

}

public function GuardarUsuario(){

$this->Modelo->id=@$\_POST['id'];

$this->Modelo->Nombres=$\_POST['Nombre'];

$this->Modelo->Apellidos=$\_POST['Apellidos'];

$this->Modelo->Telefono=$\_POST['Telefono'];

$this->Modelo->CorreoElectronico=$\_POST['CorreoE'];

$this->Modelo->Categoria=$\_POST['Categoria'];

$this->Modelo->GuardarUsuario();

$Data=$this->Modelo->ObtenerDatos();

include\_once("Vistas/Default/header.php");

include\_once("Vistas/Usuarios/index.php");

include\_once("Vistas/Default/footer.php");

}

public function ModificarUsuario($Id){

$Data=$this->Modelo->BuscarUsuario($Id);

$Categorias=$this->Modelo->ObtenerCategorias();

include\_once("Vistas/Default/header.php");

include\_once("Vistas/Usuarios/Modificar.php");

include\_once("Vistas/Default/footer.php");

}

public function BorrarUsuario($Id){

$this->Modelo->EliminarUsr($Id);

}

}

A través del siguiente código se realizan consultar a la base de datos para obtener información que será mostrara al usuario.

include("DataBase.php");

class Categoria

{

private $PDO;

public $NombreCategoria;

public $Id;

function \_\_construct()

{

$this->PDO=DataBase::Open();

}

public function ObtenerCategorias(){

try {

$sql="select \* from categoria";

$stm=$this->PDO->prepare($sql);

$stm->execute();

return $stm->fetchAll(PDO::FETCH\_OBJ);

} catch (Exception $e) {

}

}

public function GuardarCategoria(){

try {

$sql="insert into categoria(nombre)

values('".$this->NombreCategoria."')";

$stm=$this->PDO->prepare($sql);

$stm->execute();

} catch (Exception $e) {

}

}

public function ActualizarCategoria(){

try {

$sql="UPDATE appgestion.categoria SET nombre = ? WHERE categoria.id = ?";

$stm=$this->PDO->prepare($sql);

$stm->execute(array(

$this->NombreCategoria,

$this->Id

));

} catch (Exception $e) {

}

}

public function BorrarCategoria($Id){

try {

$sql="DELETE FROM appgestion.categoria WHERE categoria.id=".$Id;

$stm=$this->PDO->prepare($sql);

$stm->execute();

} catch (Exception $e) {

}

}

}