



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- Denominación del Programa de Formación: Análisis y Desarrollo de Software
- Código del Programa de Formación: 228118
- Nombre del Proyecto: Construcción de software a la medida para el control de acceso de personal, elementos de almacén y flujo de equipos externos para el centro de gestión agroempresarial del oriente.
- Fase del Proyecto: Análisis.
- Actividad de Proyecto: Especificar el modelo conceptual del sistema de información.
- Resultados de Aprendizaje Alcanzar: Elaborar el modelo conceptual de la base de datos.
- Duración de la Guía: 48

2. PRESENTACIÓN

En esta actividad de aprendizaje usted realizará la arquitectura del sistema según el patrón modelo vista controlador e implementará el patrón MVC en su proyecto formativo.

Durante la etapa de diseño del sistema una de las principales tareas es la construcción de diagramas de clases, sin embargo, pensar que el problema que se desea solucionar es único, es un error. Existen problemas comunes al desarrollo de software que han sido tipificados o estandarizados y se presentan con sus respectivas soluciones generadas a partir de diseños de interacción o interfaces. Las soluciones tipificadas, ya probadas y documentadas, se denominan patrones de diseño.

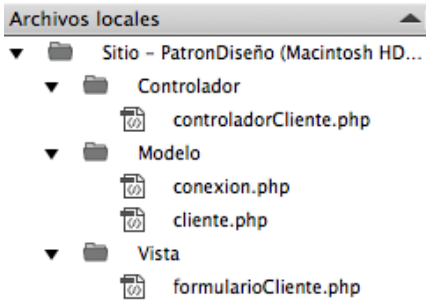
Los patrones de diseño son una herramienta que soporta la actividad de diseño de la arquitectura, proporcionando en algunas ocasiones el punto de inicio en la determinación de aspectos relacionados con la especificación de clases e interacción para el sistema de información.

La temática que se presenta en esta guía proporciona información sobre algunos de los patrones de diseño más importantes, con el fin de que sean considerados durante la fase de diseño de su proyecto formativo.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

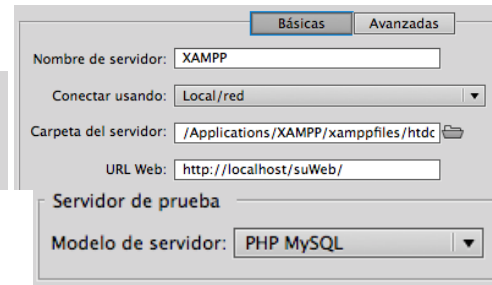
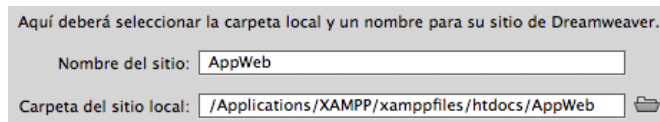
3.1 Observar el siguiente MVC creado para la entidad Cliente.

La disposición de las carpetas es la siguiente en donde podemos observar que cada una tiene el nombre del patrón de diseño (Modelo, Vista y Controlador).



3.2 Para ello debemos crear las carpetas dentro de la carpeta htdocs o http_docs.

3.3. En dreamweaver debemos agregar un sitio y configurarlo.



3.4. Creamos la clase conexión, que es la que permite conectarnos a la base de datos que ya debió crear. conexion.php (se guarda en la carpeta controlador)

```
1 <?php
2 class Conexion{
3     function conectar(){
4         $conexion = mysqli_connect("localhost","root","","supermercado");
5         mysqli_query($conexion, "SET NAMES 'utf8' ");
6         return $conexion;
7     }
8
9     function desconectar($conexion){
10        mysqli_close($conexion);
11    }
12 }
13 ?>
```

Si cree que tiene un error puede agregar en la linea 4 antes del punto y coma: **or die("Error en:".mysqli_error(\$conexion));**

3.5. Creamos la clase Cliente y probamos cada uno de sus metodos dentro de phpMyAdmin cliente.php (se guarda en la carpeta modelo)

```
1 <?php
2 class Cliente{
3
4     private $_conexion;
5     private $_idCliente;
6     private $_nombreCliente;
7     private $_documentoCliente;
8     private $_correoCliente;
9     private $_paginacion = 10;
```



```
10
11 ▼ function __construct($conexion, $idCliente, $nombreCliente, $documentoCliente, $correoCliente){
12     $this->_conexion = $conexion;
13     $this->_idCliente = $idCliente;
14     $this->_nombreCliente = $nombreCliente;
15     $this->_documentoCliente = $documentoCliente;
16     $this->_correoCliente = $correoCliente;
17 }
18
19 ▼ function __get($k){
20     return $this->$k;
21 }
22
23 ▼ function __set($k,$v){
24     $this->$k = $v;
25 }
26
27 ▼ function insertar(){
28     $insercion = mysqli_query($this->_conexion,"INSERT INTO
29 Cliente (idCliente, nombreCliente, documentoCliente, correoCliente)
30 VALUES (NULL,'$this->_nombreCliente','$this->_documentoCliente','$this->_correoCliente')");
31 $auditoria = mysqli_query($this->_conexion,"INSERT INTO
32 Auditoria (idAuditoria,detalleAuditoria,idUsuarioAuditoria,fechaAuditoria)
33 VALUES (NULL,'Inserto ".static::class.", ".$_SESSION['idUsuario'].",'CURDATE()')");
34 return $insercion;
35 }
36 ▼ function modificar(){
37     $modificacion = mysqli_query($this->_conexion,"UPDATE Cliente SET
38 nombreCliente='$this->_nombreCliente', documentoCliente='$this->_documentoCliente',
39 correoCliente='$this->_correoCliente'
40 WHERE idCliente=$this->_idCliente");
41 $auditoria = mysqli_query($this->_conexion,"INSERT INTO
42 Auditoria (idAuditoria,detalleAuditoria,idUsuarioAuditoria,fechaAuditoria)
43 VALUES (NULL,'Modifico ".static::class.", ".$_SESSION['idUsuario'].",'CURDATE()')");
44 return $modificacion;
45 }
46
47 ▼ function eliminar(){
48     $eliminacion = mysqli_query($this->_conexion,"DELETE FROM Cliente
49 WHERE idCliente=$this->_idCliente");
50 $auditoria = mysqli_query($this->_conexion,"INSERT INTO
51 Auditoria (idAuditoria,detalleAuditoria,idUsuarioAuditoria,fechaAuditoria)
52 VALUES (NULL,'Inserto ".static::class.", ".$_SESSION['idUsuario'].",'CURDATE()')");
53 return $eliminacion;
54 }
55
56
57 ▼ function cantidadPaginas(){
58     $cantidadBloques=mysqli_query($this->_conexion,
59 "SELECT CEIL(COUNT (idCliente)/$this->_paginacion) AS cantidad FROM Cliente")
60 or die(mysqli_error($this->_conexion));
61 $unRegistro=mysqli_fetch_array($cantidadBloques);
62 return $unRegistro['cantidad'];
63 }
64
65 ▼ function listar($pagina){
66     if ($pagina<=0){
67         $listado = mysqli_query($this->_conexion,"SELECT * FROM Cliente ORDER BY idCliente") or
68 die (mysqli_error($this->_conexion));
69     }else{
70         $paginacionMax = $pagina * $this->_paginacion;
71         $paginacionMin = $paginacionMax - $this->_paginacion;
72         $listado = mysqli_query($this->_conexion,"SELECT * FROM Cliente ORDER BY idCliente
73 LIMIT $paginacionMin, $paginacionMax") or die (mysqli_error($this->_conexion));
74     }
75     return $listado;
76 }
77 ?>
```

3.6 Crear la vista de la entidad del sistema de información



formularioCliente.php (se guarda en la carpeta vista)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Formulario Cliente</title>
6 </head>
7 <body>
8 <header>
9 <h1>Formulario Cliente</h1>
10 </header>
11 <table border="1">
12 <tbody>
13 <tr>
14 <th scope="col">Nombre Cliente</th>
15 <th scope="col">Documento Cliente</th>
16 <th scope="col">Email Cliente</th>
17 <th scope="col"></th>
18 </tr>
19 <?php
20 include_once("../modelo/conexion.php");
21 $objetoConexion = new conexion();
22 $conexion = $objetoConexion->conectar();
23
24 include_once("../modelo/cliente.php");
25 $objetoCliente = new cliente($conexion,0,'nombre','documento','correo');
26 $listaClientes = $objetoCliente->listar(0);
27 while($unRegistro = mysqli_fetch_array($listaClientes)){
28 echo <tr><form id="fModificarCliente".$unRegistro["idCliente"]." action="../controlador/ControladorCliente.php"
29 method="post">;
30 echo <td><input type="hidden" name="fIdCliente" value="'. $unRegistro['idCliente'].'">;
31 echo <td><input type="text" name="fNombreCliente" value="'. $unRegistro['nombreCliente'].'"></td>;
32 echo <td><input type="number" name="fDocumentoCliente" value="'. $unRegistro['documentoCliente'].'"><td>;
33 echo <td><input type="email" name="fCorreoCliente" value="'. $unRegistro['correoCliente'].'"></td>;
34 echo <td><button type="submit" name="fEnviar" value="Modificar"></button>
35 <button type="submit" name="fEnviar" value="Eliminar"></button></td>;
36 echo </form></tr>;
37 }
38 >
39 <tr><form id="fIngresarCliente" action="../controlador/ControladorCliente.php" method="post">
40 <td><input type="hidden" name="fIdCliente" value="0"></td>
41 <td><input type="text" name="fNombreCliente"></td>
42 <td><input type="number" name="fDocumentoCliente"></td>
43 <td><input type="email" name="fCorreoCliente"></td>
44 <td><button type="submit" name="fEnviar" value="Ingresar"></button>
45 <button type="reset" name="fEnviar" value="Limpiar"></button></td>
46 </form></tr>
47 </tbody>
48 </table>
49 <?php
50 mysqli_free_result($listaClientes);
51 $objetoConexion->desconectar($conexion);
52 >
53 </body>
54 </html>
```

3.7 Crear el controlador de la entidad del sistema de información:

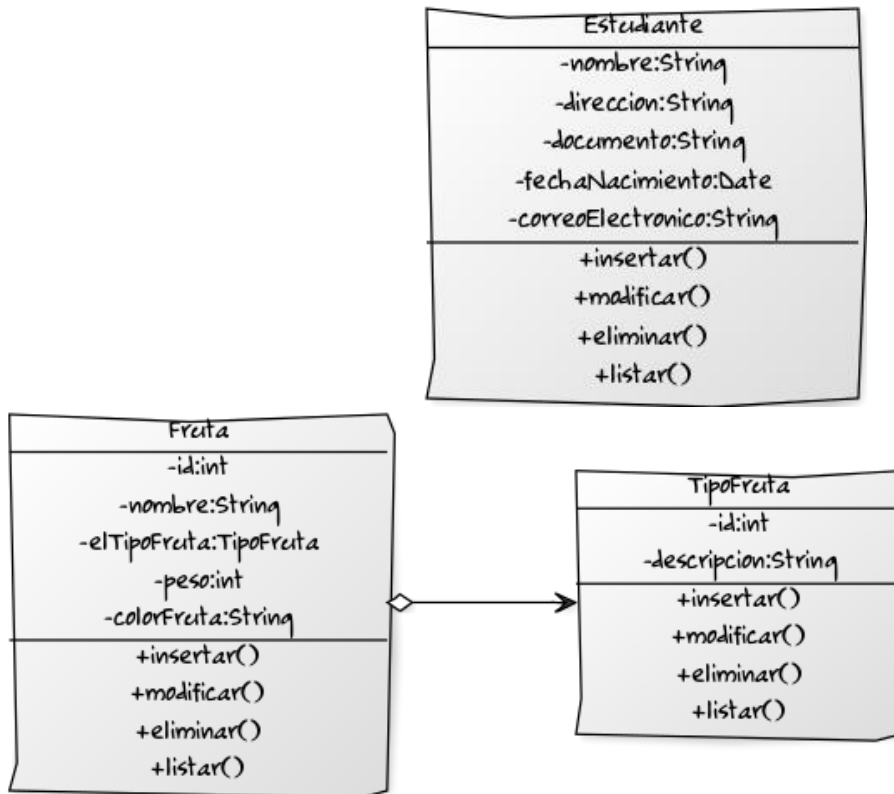
controladorCliente.php (se guarda en la carpeta controlador)

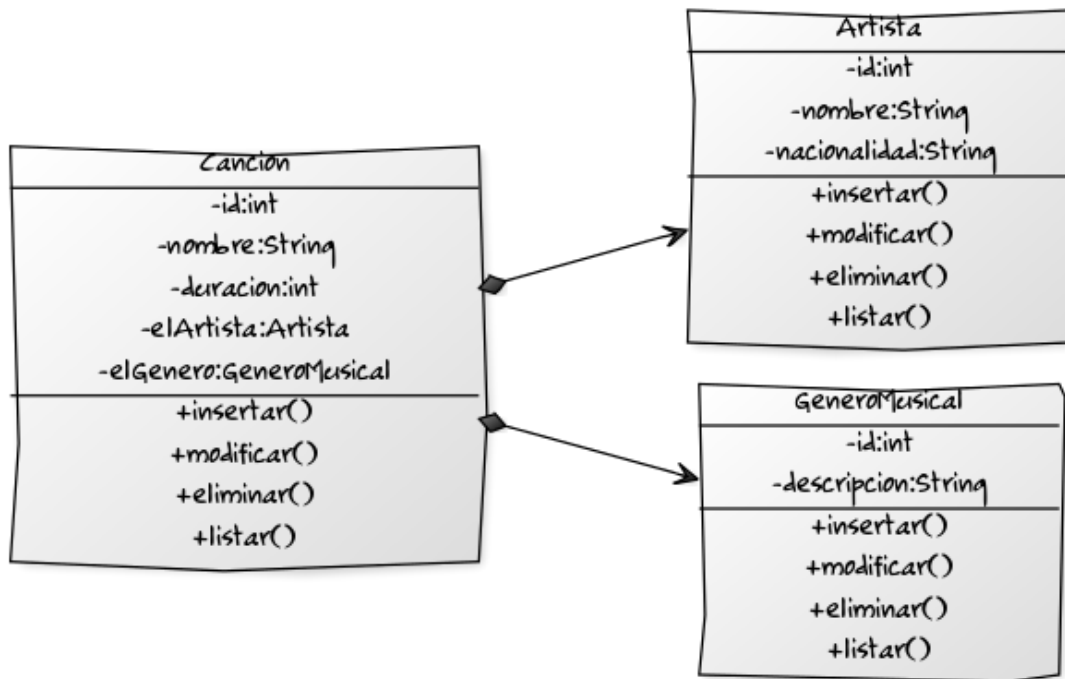
```
1 <?php
2 include_once("../modelo/conexion.php");
3 $objetoConexion = new conexion();
4 $conexion = $objetoConexion->conectar();
5
6 include_once("../modelo/cliente.php");
7
8 $opcion = $_POST["fEnviar"];
9 $idCliente = $_POST["fIdCliente"];
10 $nombreCliente = $_POST["fNombreCliente"];
11 $documentoCliente = $_POST["fDocumentoCliente"];
12 $correoCliente = $_POST["fCorreoCliente"];
13
14 $nombreCliente = htmlspecialchars($nombreCliente);
15 $documentoCliente = htmlspecialchars($documentoCliente);
16 $correoCliente = htmlspecialchars($correoCliente);
17
18 $objetoCliente = new Cliente($conexion, $idCliente, $nombreCliente, $documentoCliente, $correoCliente);
19
```

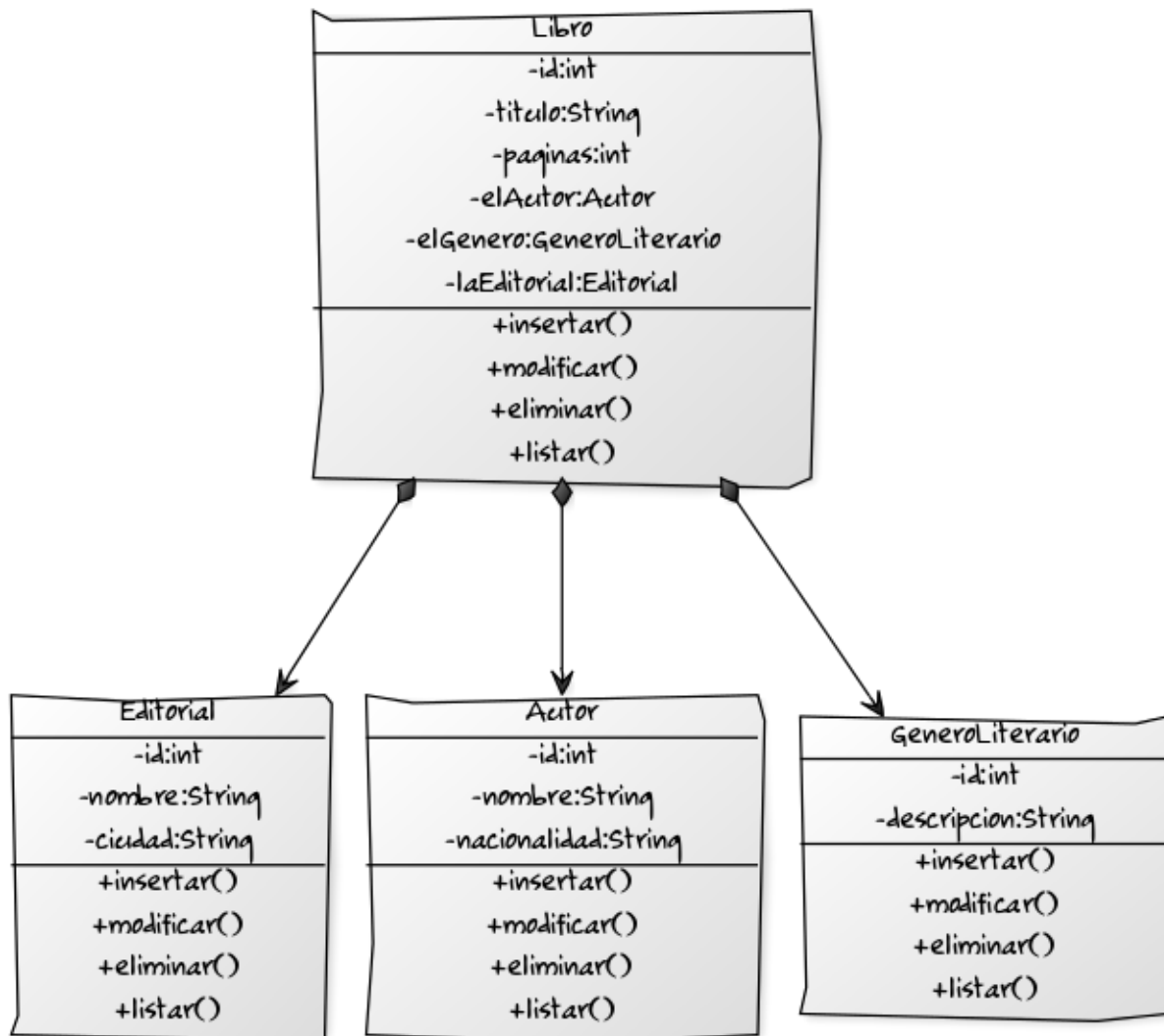


```
20 switch($opcion){
21     case 'Ingresar':
22         $objetoCliente->insertar();
23         $mensaje = "Ingresado";
24     break;
25     case 'Modificar':
26         $objetoCliente->modificar();
27         $mensaje = "Modificado";
28     break;
29     case 'Eliminar':
30         $objetoCliente->eliminar();
31         $mensaje = "Eliminado";
32     break;
33 }
34
35 $objetoConexion->desconectar($conexion);|
36 header("location:../vista/formularioCliente.php?msj=$mensaje");
37 ?>
```

3.8 Realizar MVC para las entidades:







3.9 Realizar un sistema de información para llevar el control de citas en consultorio odontológico, que se puede registrar pacientes, odontólogos, consultorios y citas.

3.10 Realizar un sistema de información para llevar el control de los cliente de un hotel, que se puedan registrar clientes, habitaciones, checkin, checkout, recepcionistas, mucamas

Suba todos los archivos generados al link designado por el instructor en la plataforma LMS.

Para desarrollar la anteriores actividades se requiere de un ambiente tipo aula con mesas y equipos de computo con conexión a internet, UPS, sillas ergonómicas, televisor con entrada HDMI o MHL, tablero acrílico, extintor blanco, escoba, recogedor, papeleras de reciclaje.

Para desarrollar la anterior actividad se requiere de los siguientes materiales: Tener en el computador instalado cualquier sistema operativo con interfaz gráfica (Microsoft windows, linux, macos), navegadores



web: firefox, google chrome, opera; IDEs como visual studio code, brackets.io, notepad++, netbeans y servidores como xampp o easyphp, suite ofimatica libreoffice o microsoft office.

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : Evidencias de Desempeño Evidencias de Producto: - MVC de los ejercicios - MVC del Sistema de información en PHP	Elabora el prototipo, representando la funcionalidad del sistema de información, para dar respuesta a las necesidades de la organización, según protocolos de diseño.	Técnica de Evaluación: Valoración de producto Instrumento de Evaluación: Lista de Verificación

5. GLOSARIO DE TÉRMINOS

MVC: Modelo Vista Controlador

Patrón: Modelo o guía a seguir.

Patrón de diseño: Conjunto de técnicas y métodos realizados y ya probados para solucionar problemas comunes encontrados al momento de diseñar un sistema de información.

6. REFERENTES BIBLIOGRÁFICOS

Java Desing Patterns. Sarcar Vaskaran. Apress. 2016

http://wiki.hashphp.org/PDO_Tutorial_for_MySQL_Developers..

Professional ASP.NET Desing Patterns. Scott Millett. Wrox. 2010

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Germán Alberto Cruz Vargas	Instructor	Coordinación Académica	15 de febrero 2017

8. CONTROL DE CAMBIOS



	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)	Miguel Ángel Tejedor Mendoza	Instructor	Coordinación Académica	10 de febrero 2023	Se actualizó al formato actual.