

# PROGRAMACIÓN DE COMPONENTES

## WEB OPEN SOURCE - EVALUACIÓN 2

### Competencia asociada:

Desarrollar back end de aplicaciones utilizando estándares de la industria

### Unidad de Competencia:

1. Construye componentes que consumen servicios interoperables de acuerdo a los estándares de la industria.
2. Construye componentes que disponibilizan servicios interoperables de acuerdo a los estándares de la industria y aplicando políticas de seguridad.
3. Construye documentos con estructuras de información interoperables utilizando los estándares de la industria.

### Consideraciones importantes

1. La evaluación es un encargo.
2. Usted posee un periodo en envío limitado para poder realizar el encargo. Luego de ello se entrega la retroalimentación de su código
3. La copia o plagio de algún encargo y/o evaluación será sancionada con nota 1.

### Enunciado

El dueño del popular hotel “Hotel Miras Del Mar”, le contacta para solicitar la creación de un sistema web que permita consultar los precios de sus habitaciones.

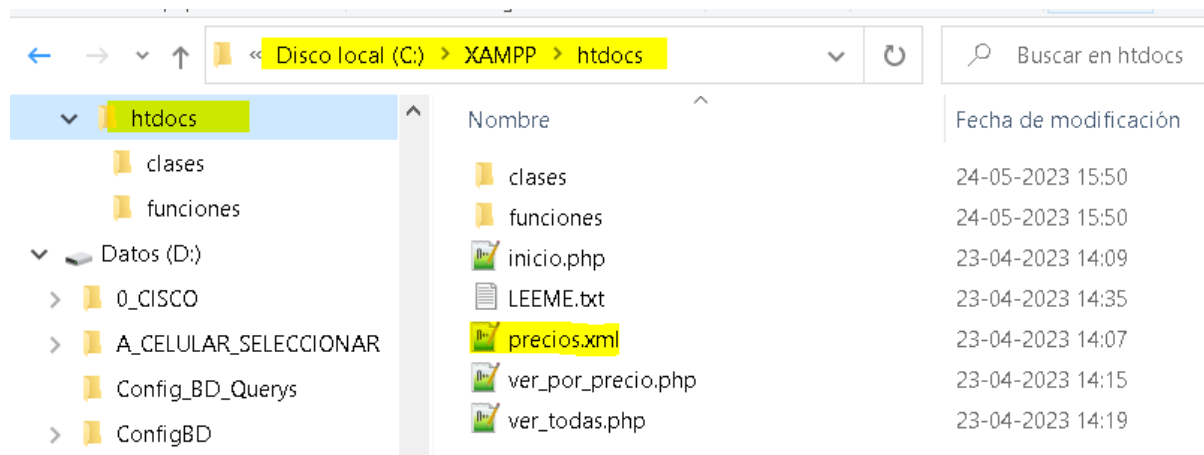
El objetivo es realizar las consultas con y sin filtros.

Debe considerar las siguientes indicaciones:

- Todos los precios **deben cargarse** desde un archivo XML (genere archivo XML **sólo con los valores de precios**). El archivo debe llamarse “**precios.xml**” (y estar en la carpeta principal del proyecto; C:\XAMP\htdocs) allí estarán **sólo** los precios relacionados con las habitaciones. Estos **deben ser sólo 4**: 50000, 45000, 40000, 35000; **no use más tags con otros datos**.
- A nivel de base de datos, una habitación se compone de un código, un nombre y precio relacionado
- Las consultas deben ser realizadas a una base de datos MySQL llamada **hotelmdbm**
- **Debe utilizar la extensión PDO** para realizar las consultas a la base de datos

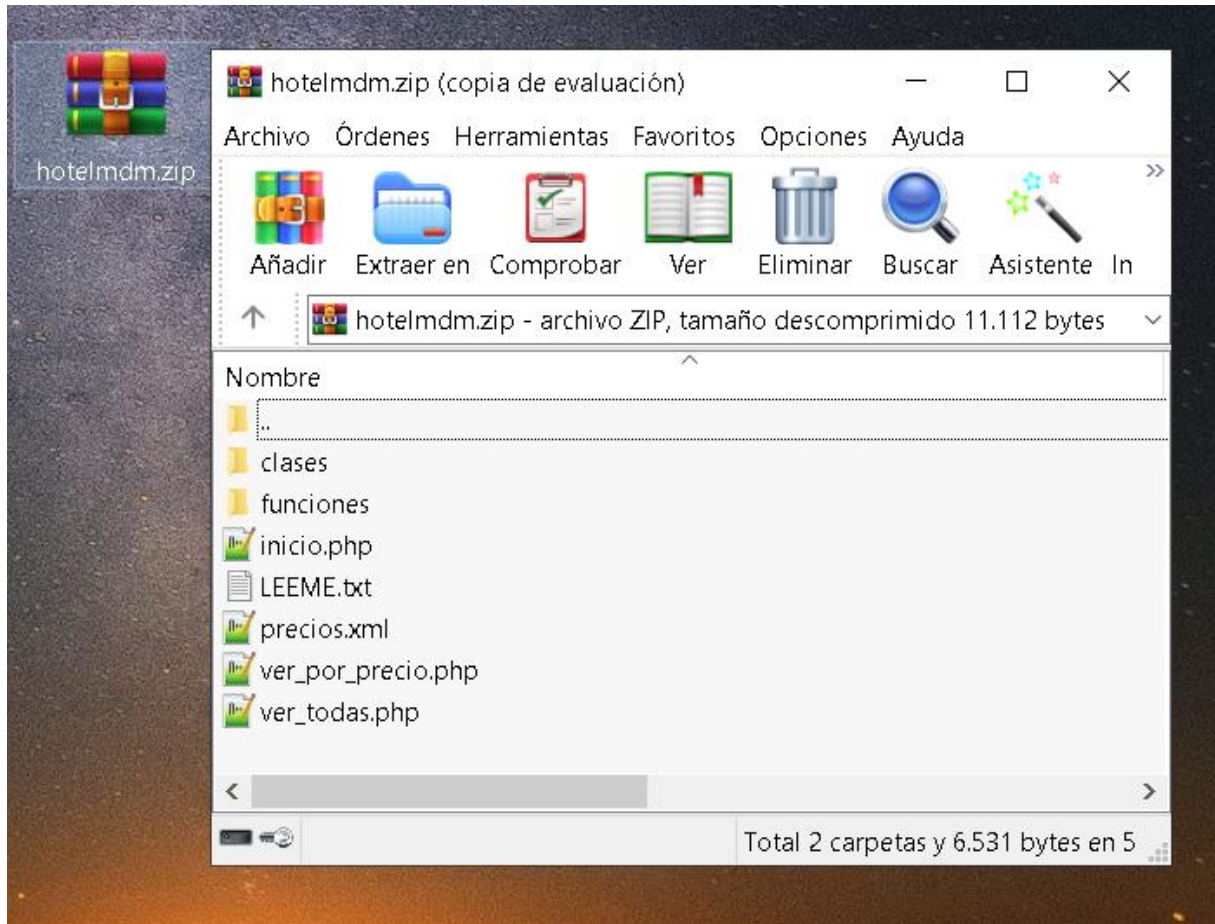
## Paso 1

Este proyecto debe estar en la carpeta principal de alojamiento, es decir, si utiliza XAMPP, debe dejar todo su proyecto en la carpeta C:\XAMPP\htdocs; debería verse así:



Allí debe crear las subcarpetas “clases” y “funciones”, para guardar los archivos correspondientes de forma ordenada, tendiendo a usar el modelo MVC.

Por otra parte, cuando suba su proyecto (en ZIP o RAR) debe verse de la siguiente manera:



En este caso se ha llamado “hotelmdm.zip”; pero este nombre no tiene mayor importancia, pues al bajar (para revisar), la plataforma renombrará este archivo con su nombre completo.

La base de datos “hotelmdm” tiene sólo la tabla “**habitacion**” con las siguientes especificaciones:

- codigo (int (10) – **Llave primaria** – No Nulo)
- precio (int (10) – No Nulo)
- nombre (varchar(50)) – No Nulo

Con el objeto de concentrarse en la solución, se le comparte el siguiente script (cópelo en un archivo .SQL y ejecútelo en el MySql de su equipo, para cargar la BD y así utilizarla directamente – Puede usar phpmyadmin):

```
DROP DATABASE IF EXISTS hotelmdm;

CREATE DATABASE IF NOT EXISTS hotelmdm;

USE hotelmdm;

CREATE TABLE habitacion (

    codigo int (10) NOT NULL,

    precio int (10) NOT NULL,

    nombre varchar (50) NOT NULL,

    PRIMARY KEY (codigo)

);

Insert into habitacion values ('1','35000','Single');

Insert into habitacion values ('2','40000','Single Plus');

Insert into habitacion values ('3','45000','Doble');

Insert into habitacion values ('4','50000','King Size');
```

**Nota:** Este ítem no será evaluado

La tabla viene con datos de prueba. Los precios de la columna “precio” **están relacionados** con los precios del archivo XML.

## Paso 2

Crear la clase “**conexion**” con un atributo llamado “**conexion**”. En el constructor debe crear la conexión y guardarla en el atributo “conexion”. Solo se debe crear el método accesor para el atributo “conexion”.

Crear la clase “**habitacion**” para representar la cantidad habitación. Los atributos de la clase “habitacion” **deben ser los mismos** que las columnas de la tabla “habitación”.

Es decir:

columna “codigo” -> atributo “codigo”

columna “precio” -> atributo “precio”

columna “nombre” -> atributo “nombre”

## Paso 3

Crear los siguientes archivos (todos deben quedar en la carpeta funciones):

- BuscarPorPrecio.php: Debe crear una función que realice una consulta a la tabla “habitacion”, y filtrar por precio. El usuario es quien seleccionará el precio. Los campos por mostrar son: “codigo”, “nombre” y “precio”. La función **debe retornar un array** con las habitaciones encontradas.
- BuscarTodas.php: Debe crear una función que realice una consulta sin filtros a la tabla “habitacion”. Los campos por mostrar son: “codigo”, “nombre” y “precio”. La función **debe retornar un array** con las habitaciones encontradas.
- CargarPrecios.php: Debe crear una función que realice la carga de los datos contenidos en el archivo “precios.xml”, **retornando un array con los precios encontrados**.

## Paso 4

Debe crear las siguientes páginas (en la raíz del proyecto):

**“inicio.php”:** Es el inicio del sistema y debe tener un formulario con un select (combobox) con los precios cargados desde el archivo “precios.xml”, para ello debe:

- Invocar “CargarPrecios.php” como biblioteca.
- Ejecutar la función, guardando el resultado en una variable de tipo “array”.
- Procesar el arreglo para generar el select.

El formulario debe llamar al archivo “ver\_por\_precio.php” (está en la misma carpeta que “inicio.php”). Cuando el usuario selecciona un precio (desde el select), debe consultar por este valor como filtro, en la tabla “habitacion” y allí mostrar todos los datos (codigo, nombre y precio), para ello debe:

- Invocar “BuscarPorPrecio.php” como biblioteca.
- Ejecutar la función, guardando el resultado en una variable de tipo “array”.
- Procesar el arreglo para generar para desplegar los datos

En inicio.php, también **debe mostrar un link** que permita ver todas las habitaciones con sus datos correspondientes; este link apunta a “ver\_todas.php”, como se observa en la siguiente imagen de ejemplo:

---

Seleccione precio

50000 ▼
50000
45000
40000
35000

[Buscar todas las habitaciones](#)

Al usar el link “ver\_todas.php”, debe mostrar como resultado todos los datos: código, nombre y precio; de todos los registros de la base de datos, como se observa en la siguiente imagen de ejemplo:

Código: 4444

Nombre: 104

Precio: 45000

Código: 5555

Nombre: 105

Precio: 45000

**OBSERVACIONES / TIPS:**

- 1) Es importante comprender que el sistema debe ser “portado” desde un equipo a otro; por tanto, no debe presentar problemas en este proceso (es parte de la evaluación). En ambientes reales, esto ocurre constantemente, puesto que se desarrolla y prueba en entornos de Desarrollo/QA para finalmente “portar” hacia el entorno de “producción”. En este caso, su equipo sería “Desarrollo/QA” y donde se revisará su sistema (equipo del profesor) será el entorno final de producción. Para asegurar este punto debe tener en cuenta los siguientes puntos:
  - a. Debe ser estricto en el uso de mayúsculas y minúsculas en los nombres de archivo, pues estos desarrollos, generalmente se utilizan en servidores que son sensibles a esta diferencia.
  - b. **Debe usar rutas relativas**, evitar rutas completas y en formato “Windows”. Una buena forma de probar es crear una copia dentro de su proyecto (en una carpeta temporal) y probar que funcione 100% dentro de esta copia (recuerde borrar esta copia al momento de comprimir su entrega final).

**Ejemplo de rutas Windows a evitar en su código es:**

**C:\xampp\htdocs\...**

- 2) No se usó “index.php” a propósito, para tener una mayor flexibilidad al momento de revisar el código.
- 3) Se aconseja ajustarse a lo solicitado en las instrucciones y lo expuesto en la rúbrica de revisión.
- 4) Para crear una variable de tipo array, debe utilizar la siguiente instrucción:  

```
$arrElementos = array();
```
- 5) Para agregar elementos a un array, debe utilizar la siguiente instrucción:  

```
array_push($arrElementos, $nuevoElemento);
```
- 6) Para efectos de la revisión, asegure que el acceso a la BD desde PHP, siempre use como usuario root y como password la cadena vacía.



- 7) Si usted encuentra necesario hacer indicaciones especiales para la carga de su proyecto, genere un archivo llamado LEEME.txt en la carpeta principal de su proyecto con las explicaciones que estime pertinente.
- 8) Esta asignatura, como se indica en la competencia asociada, se centra en el “backend” de las aplicaciones, por lo cual, **debe utilizar solamente las siguientes tecnologías:** HTML, PHP, MySQL y JavaScript básico (como por ejemplo mensajes con “alert”). Adicionalmente, para la simplicidad en la revisión **no debe usar otros elementos que escapen al contexto de la asignatura como CSS, JQuery, AJAX u otras tecnologías más recientes y orientadas al frontend (de lo contrario tendrá descuentos).** Cabe señalar que lo anterior, si se da en contextos de proyecto reales, es decir, estas tecnologías sí se mezclan.

## Criterios de evaluación

### Escala de apreciación

INDICADORES DE LOGRO	PUNTAJE TOTAL 100 PUNTOS
Comprime archivos según lo señalado en paso 1 del enunciado	2
Genera estructura de carpetas de acuerdo con los requerimientos	2
Crea archivo XML con los datos señalados	8
Crea clase conexion para guardar los eventos de conexión	6
Crea clase habitacion según requerimiento (3 getters y 3 setters = 6)	12
Declara funciones de acuerdo con los requerimientos (3 funciones)	12
Crea la página inicio de acuerdo con los requerimientos.	8
Extrae los precios del XML y los almacena en un arreglo	8
Genera select con valores cargados desde el XML	8
Crea BuscarPorPrecio como biblioteca con función "llamable"	6
Crea BuscarTodas como biblioteca con función "llamable"	6
Crea CargarPrecios como biblioteca con función "llamable"	6
Crea la página ver_por_precio, carga biblioteca y procesa arreglo.	8
Crea la página ver_todas, carga biblioteca y procesa arreglo	8
<b>PUNTAJE TOTAL</b>	<b>100</b>

## AUTOEVALUACIÓN

Estimado estudiante:

La autoevaluación es la valoración y/o reflexión que usted realiza sobre su proceso de aprendizaje. Este ejercicio es fundamental para identificar tanto debilidades como fortalezas de la evaluación realizada.

Para realizar la autoevaluación usted deberá contestar las siguientes preguntas, las cuales se encontrarán en la pestaña **AUTOEVALUACIÓN**, donde deberá completar cada una de ellas. *(No se deberán contestar en esta planilla)*

Conteste con toda la sinceridad posible, ya que será un insumo para la retroalimentación que realizará posteriormente el docente.

- ¿Qué hice bien?
- ¿En qué necesito ayuda?
- ¿Sobre qué quiero saber más?
- ¿Qué aprendí en esta evaluación?
- ¿Qué debo mejorar?
- ¿Qué acciones realizaré para mejorar?