

PROGRAMACIÓN DE COMPONENTES

WEB OPEN SOURCE - EXAMEN

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 restaurant “Reservoir”, le ha encargado generar un software que tiene como prioridad controlar la cola de pedidos de las mesas. Requiere de una web que permita registrar los pedidos de las mesas y luego mostrarlas en una pantalla para que el cocinero y su equipo logren atender a los clientes.

Se pueden reutilizar partes de los códigos anteriores, pero debe comprenderse que este es una app totalmente distinta a la anterior; por ende debe adaptar si es el caso.

La experiencia parte por un menú en donde puede elegir entre ir a los pedidos o ver la producción (este archivo se llama "inicio.php")

Menu

[Ir a Pedidos](#)

[Ver Produccion](#)

Estando en "ir a pedidos", si hace clic en Ir a Pedidos se debe mostrar el siguiente formulario

Pedido

Mesa	<input type="text" value="1"/>
Descripcion	<input type="text"/>
<input type="button" value="Agregar"/> <input type="button" value="Ver pedido"/> <input type="button" value="Volver"/>	

La mesa simplemente son números consecutivos del 1 al 20 (**se debe** usar un fragmento de PHP para generar estos 20 elementos en el select)

La descripción es el texto que ingresa el mesero (petición del cliente)

Si hace clic en agregar, debe mostrar un mensaje de pedido agregado (podría ser con un alert de javascript) o un mensaje de error si no se pudo grabar (según corresponda) y luego "resetear" o "recargar" el formulario, para poder seguir agregando si se desea.

Si hace clic en “Ver Pedido”, debe mostrar el siguiente formulario:

Ver Pedido

Mesa	Descripcion	Accion
1	Pizza Italiana Familiar Doble Queso	Eliminar
2	Entrada Ensalada Cesar	Eliminar

[Volver](#)

Si hace click en “volver” debe regresar a la pantalla anterior, esto es “ir a pedidos” con el formulario respectivo.

Si hace click en eliminar, se debe borrar ese ítem del pedido, es decir, de la base de datos.

Si por algún motivo no hay pedidos registrados en la BD (por ejemplo, se borraron todos); al momento de pulsar “Ver Pedido” **debe indicarlo mediante un mensaje apropiado.**

Si desde el menú principal “inicio.php”, hace click en “Ver Producción” se muestra el siguiente formulario:

Ver Produccion

Mesa	Descripcion	Accion
1	Pizza Italiana Familiar Doble Queso	Finalizar
2	Entrada Ensalada Cesar	Finalizar

[Volver](#)

Si hace click en “volver” debe regresar a la pantalla anterior, esto es “inicio.php”

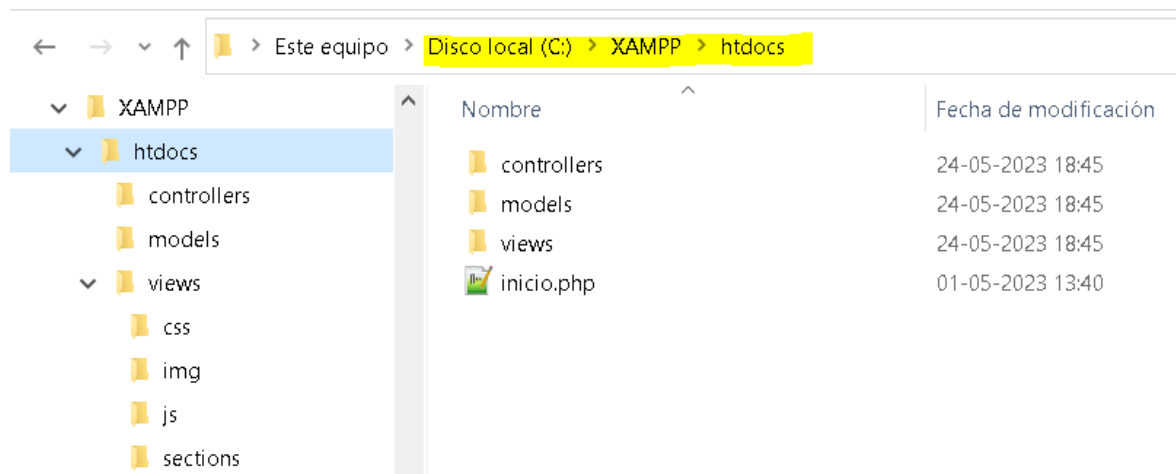
En esta pantalla (ver producción) se deben mostrar todos los pedidos que actualmente están vigentes. Si hace clic en finalizar (es un link), entonces ese

pedido sale de producción, cambiando su estado a finalizado. Se entenderá estado=0 como pedido creado y estado=1 como pedido finalizado.

Si por algún motivo, no hay pedidos registrados en la BD, debe indicarlo mediante un mensaje apropiado.

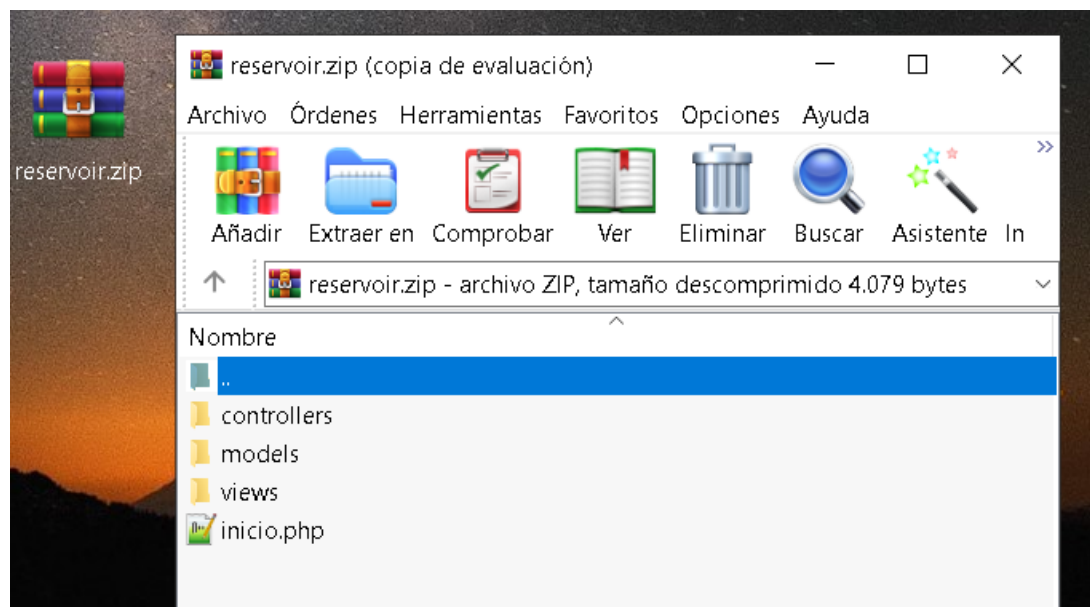
Cabe destacar que tanto en “ver producción” como en ver pedidos, no deben verse los registros ya finalizados, eso quedará pendiente para una futura funcionalidad, que permita buscar pedidos finalizados a modo de historial.

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:



Debe contener las subcarpetas “models”, “views”, “controllers”; al igual que el proyecto anterior. Dentro de la carpeta “views” debe crear las carpetas “css”, “img”, “js” y “sections”. (para un supuesto uso futuro).

Cuando suba su proyecto (en ZIP o RAR) debe verse de la siguiente manera:



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

En “sections” debe **reusar y modificar** los archivos que representan la cabecera (header.html) y pie (footer.html) del proyecto anterior.

Modelo de datos

La BD “**reservoir**” tiene sólo la tabla “**pedido**” con las siguiente estructura:

id: int(10) – **Llave primaria** – No nulo – **Autoincremental**

mesa: int(2) – No nulo

descripcion: Varchar(128) – No nulo

estado: int(2) – No nulo. Se deben usar valores para estado, por ejemplo 0=iniciado y 1=finalizado

fechahora: Almacena la fecha y hora de registro del pedido

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 reservoir;

CREATE DATABASE IF NOT EXISTS reservoir;

USE reservoir;

CREATE TABLE pedido (

    id int (10) AUTO_INCREMENT NOT NULL,

    mesa int (2) NOT NULL,

    descripcion varchar (128) NOT NULL,

    estado int (2) NOT NULL,

    fechahora datetime NOT NULL,

    PRIMARY KEY (id)

);
```

Nota: Este ítem no será evaluado

Dentro de la carpeta raíz del proyecto, debe crear un archivo llamado “inicio.php”; por lo que se recomienda que en esa misma carpeta, genere 2 archivos “pedidos.php” y “producción.php” enlazados desde “inicio.php” y cada uno debe:

- Cargue el controlador
- Cree una nueva instancia del controlador pedido
- Ejecute el método respectivo donde y cuando corresponda: crear, ver_pedido, borrar, ver_produccion, finalizar.

De este modo, se asegura que se usan clases, métodos, etc, es decir, en definitiva, se aplica el modelo MVC

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 efectos de la revisión, asegure que el acceso a la BD desde PHP, siempre use como usuario root y como password la cadena vacía.
- 5) 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.
- 6) 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 escapan 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
Crea la estructura de carpetas según las instrucciones (MVC / sitio)	4
Crea clase conexión para guardar los eventos de conexión	6
Crea clase Pedido para guardar datos de cada mesa (5 setters; 5getters)	10
Crea clase controladora de Pedido (5 métodos)	20
Crea en la carpeta views el formulario de ingreso de pedido	4
El formulario de pedido logra agregar	4
Muestra mensaje apropiado al agregar pedido	4
Ver pedido logra ver los pedidos de las mesas	8
Ver pedido logra eliminar un pedido	8
Indica mensaje cuando no hay pedidos	4
Volver (desde ver pedidos) permite ir a la pantalla anterior (ir a pedidos)	2
Volver (desde ir a pedidos) permite ir a la pantalla anterior (inicio)	2
Ver Produccion, logra identificar diferentes los pedidos con estado apropiado (no debe mostrar los ya finalizados)	8
Ver Producción logra finalizar un pedido	8
Indica mensaje cuando no hay pedidos para finalizar	4
Volver (desde ir a producción) permite ir a la pantalla anterior (inicio)	2
PUNTAJE TOTAL	100

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?