MEMORIA

PRÁCTICA 2: PHP y MySQL.

Diagrama, Esquemático

Descripción generada automáticamente

DESARROLLO DE APLICACIONES EN INTERNET (2806)

Autor: Luis David Diaz Mesa

Profesor/es: Juan Pedro Moreno Martin-Viveros

Antonio Peñalver Benavent

INDICE

[Introducción 2](#_Toc137414997)

[Parte 1: Desarrollo de la interfaz gráfica 2](#_Toc137414998)

[Autenticación 2](#_Toc137414999)

[Gestión de clientes, embarcaciones y repuestos 3](#_Toc137415000)

[Gestión de facturas 3](#_Toc137415001)

[Parte 2: Desarrollo de la interfaz Consultas 4](#_Toc137415002)

[Parte 3: MVC con Laravel 4](#_Toc137415003)

# Introducción

Esta aplicación web desarrollada con PHP, hemos creado los módulos necesarios para cada tabla de una base de datos con MySQL, usando para ello XAMPP, y de esta forma insertar, borrar, consultar y actualizar los datos de los diferentes módulos.

Por otra parte, hemos hecho el apartado de Laravel usando el sistema de modelo-vista-controlador. También se ha empleado el ORM Eloquent para interactuar con la base de datos.

**\*Para algunas consultas con la base de datos se ha tenido que cambiar el nombre del campo “E-mail” a “Email”, por ejemplo, en la tabla clientes.**

# Parte 1: Desarrollo de la interfaz gráfica

Menú principal.

*Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente*

## 

## Autenticación

A través de una página de inicio sesión y de las sesiones en php, controlamos el acceso a la aplicación, de tal modo que al iniciar sesión correctamente se crea una sesión en php “isAuth” que le damos un valor de *“true”*, después de este paso redireccionamos al usuario al menú principal de la aplicación.

Para controlar el acceso a las páginas de la aplicación se añade al principio de cada fichero un include del fichero “auth.php” que verifica que la variable de sesión “isAuth” tenga el valor de *“true”*, en caso contrario direcciona al usuario a una página que le informa que tiene no tiene acceso a la página.

## Gestión de clientes, embarcaciones y repuestos

Estos 3 módulos siguen una estructura bastante igual, ofrecen las mismas operaciones sobre sus respectivas tablas, la mayoría de los datos se pasan mediante el método POST a los archivos (o sobre el mismo archivo) donde se obtienen los datos de la tabla correspondiente de la BD y se pintan en el cliente.

Vista del módulo gestión de cliente

*Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media*

## Gestión de facturas

Este módulo un poco más complicado que los anteriores, se ha tenido que desarrollar obteniendo los datos de múltiples tablas, por lo que se tuvo que instanciar en varias ocasiones el objeto de conexión con la BD. Además, para añadir las líneas de detalle he usado un script de JavaScript con php dentro. Las funcionalidades que ofrece este módulo son las mismas que las anteriores pero un poco más complicadas de implementarlas ya que hay que tener en cuenta varias cosas, como de que tabla obtener los datos, o calcular ciertos campos.

Vista de la función Añadir Factura del módulo gestión de facturas

*Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente*

# Parte 2: Desarrollo de la interfaz Consultas

Para este modulo ofrecemos dos tablas sobre las cuales se le ofrece al usuario una serie de campos a rellenar de forma opcional para obtener los registros de la tabla de la BD relacionados con los campos rellenados, y en caso de no rellenar ninguno se devuelven todos los registros de la tabla.

Para algunos campos se ofrecen una serie de opciones a elegir a modo de restricción.

Vista de consulta sobre la tabla factura (ningún campo pasado)

*Tabla

Descripción generada automáticamente*

# Parte 3: MVC con Laravel

Para esta parte se han creado dos rutas las cuales se le pasa la URL solicitada con los parámetros a consultar en la base de datos.

Para ello debemos haber creado dos modelos de tablas Cliente y Embarcación con artisan, y por consiguiente los ficheros controladores para cada uno, que son los que se encargan de realizar las consultas necesarias mediante el ORM Eloquent a la base de datos, para luego pasarlos a la vista correspondiente.

En la vista recibimos el objeto con los datos obtenidos y mediante un bucle foreach lo pintamos en el cliente.

Código de rutas.

*Interfaz de usuario gráfica, Texto

Descripción generada automáticamente*

Ejemplo de controlador para la funcionalidad de actualizar el año de una embarcación.

*Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente*

Ejemplo de vista que devuelve el controlador anterior.

Texto

Descripción generada automáticamente

Ejemplo de vista de resultado de la primera ruta

*Tabla

Descripción generada automáticamente*

Ejemplo de vista de resultado de la primera ruta

Tabla

Descripción generada automáticamente