

Asignatura	Datos del alumno	Fecha
Computación Cliente y Servidor	Apellidos:	
	Nombre:	

UNIR

Creación de un sitio web con PHP y JavaScript

Computación Cliente y Servidor

Raúl López Blanco
Luis Emilio Velasco Sánchez
Cristhian Michel Castillo García

19 de enero de 2021

Asignatura	Datos del alumno	Fecha
Computación Cliente y Servidor	Apellidos:	
	Nombre:	

Índice

Introducción	3
Tecnologías utilizadas	3
Componentes	3
Clases de Datos	7
Implementación del Modelo Vista Controlador	7
Complemento Ajax en las vistas	8

Asignatura	Datos del alumno	Fecha
Computación Cliente y Servidor	Apellidos:	
	Nombre:	

Introducción

Este sitio web tiene el propósito de que los usuarios que se registren en el sitio puedan añadir tuits una timeline común para todos ellos de forma que todos puedan leer las aportaciones anteriores. Para su mejor interacción y visualización la página se ha servido en Heroku, una plataforma “as a service” que nos permite desplegar, correr e interactuar con nuestros desarrollos.

Tecnologías utilizadas

Para la implementación de este sitio web se ha utilizado Lumen un framework de PHP así como el motor de plantillas de Laravel que ha permitido desarrollar código de una manera más ágil manteniendo el patrón modelo vista controlador (MVC)

Para la base de datos se ha utilizado mysql.

Se ha implementado el patrón de diseño Repository para la capa de datos.

Ejecución del código

Las instrucciones para la ejecución del código se encuentran en el archivo README.md

Componentes

El sitio está compuesto por:

- Un formulario de registro: Recoge el nombre, nombre de usuario y contraseña y permite que el usuario se cree una cuenta. Realiza una petición AJAX cuando se pulsa el botón de registrarse.

Asignatura	Datos del alumno	Fecha
Computación Cliente y Servidor	Apellidos:	
	Nombre:	

Iniciar sesión

Registro

Nombre

Nombre de usuario

Contraseña

Registrarse

Asignatura	Datos del alumno	Fecha
Computación Cliente y Servidor	Apellidos:	
	Nombre:	

- Un formulario de login: Recoge un nombre de usuario y una contraseña y realiza una llamada asíncrona a la base de datos para comprobar si los datos introducidos son correctos o no. Realiza una llamada de AJAX cuando se pulsa el botón de Iniciar Sesión


The image shows a login form interface. At the top right, there is a link labeled "Registrarse" in blue. Below it, the title "Iniciar sesión" is displayed in bold black text. The form contains two input fields: "Nombre de usuario" and "Contraseña". Below these fields is a large blue button labeled "Iniciar sesión".

- Una vista de tuits: Permite visualizar los tuits del resto de los usuarios almacenados en la base de datos, estos son recargados de manera asíncrona cada vez que tuiteamos y cada cuatro segundos para ver tuits nuevos sin la necesidad de twittear.

Asignatura	Datos del alumno	Fecha
Computación Cliente y Servidor	Apellidos:	
	Nombre:	


Inicio

Cerrar sesión




¿Qué está pasando?

Tweet




Raul @raulb34 · hace 1 hora

Esperemos que no nieve




Luis Emilio Velasco Sanchez @emibloque · hace 6 horas

Mañana parece que va a llover




Jack Dorsey @jack · hace 6 horas

Cool social network ❤️



Luis Emilio Velasco Sanchez @emibloque · hace 7 horas

El segundo



Luis Emilio Velasco Sanchez @emibloque · hace 22 horas

El primer Tweet!

Asignatura	Datos del alumno	Fecha
Computación Cliente y Servidor	Apellidos:	
	Nombre:	

Clases de Datos

Las clases de datos se encuentran en el directorio `/app/Repositories/` y son la siguientes:

- a) DBConnector: Clase que define el objeto de conexión con Maria db, con 3 métodos:
 - a. Select: Selecciona los registros ejecutando el query que se le pasa por parámetro.
 - b. Insert: Inserta los objetos mediante el query que se le pasa por parámetro.
 - c. getConnection: Crea el objeto de conexión con los datos del archivo de entorno cargado por el framework Laravel.
- b) TweetRepository: Repositorio para la entidad Tweet, con 2 métodos:
 - a. findAll: obtiene todas las entidades guardadas en la tabla tweets.
 - b. Save: guarda un tweet en la tabla.
- c) UserRepository: Repositorio para la entidad Usuario con 2 métodos:
 - a. Findbyusername: ejecuta un query para obtener la entidad de un usuario mediante su nombre de usuario. Utilizado en el login.
 - b. Save: guarda un nuevo usuario. Utilizado durante el registro de usuarios.

Implementación del Modelo Vista Controlador

Ubicación de los modelos: `/app/Models`

- a) Tweet
- b) User

Ubicación de los controladores: `/app/Http/Controllers`

- a) Auth/LoginController
- b) Auth/RegisterController

Asignatura	Datos del alumno	Fecha
Computación Cliente y Servidor	Apellidos:	
	Nombre:	

c) FeedController

Ubicación de las vistas /resources/views

- a) Auth/login.blade
- b) Auth/register.blade
- c) Feed/index.blade
- d) Layouts/app.blade

Complemento Ajax en las vistas

Vista Feed de tweets

Método sendTweet: El método envía una solicitud POST al controlador de tweets y este se guarda en base de datos.

Método fetchtweets: El método se ejecuta cada 4 segundos y obtiene la lista de tweets que se hayan insertado en la tabla, se ejecuta de manera asíncrona para permitir una carga de los tweets que se hayan insertado fuera del navegador del usuario actual.

Vista Registro:

Método register: ejecuta una solicitud post que envía los datos de un usuario nuevo al controlado de usuarios para que este lo ingrese en la base de datos.