UNIVERSIDAD DE SAN CARLOS DE GUATEMALA CENTRO UNIVERSITARIO DE OCCIDENTE DIVISIÓN DE CIENCIAS DE LA INGENIERÍA INTRODUCCIÓN A LA PROGRAMACION 2



MANUAL TÉCNICO

1. Introducción: Este manual técnico describe el desarrollo de un sistema web que integra múltiples tecnologías, incluyendo Java, Angular, SQL, y MySQL, con el fin de permitir a los usuarios gestionar anuncios en línea. El sistema permite a los anunciantes crear y publicar anuncios en diferentes formatos (texto, imágenes, y videos), verificar su saldo disponible y realizar pagos por el servicio.

2. Tecnologías Utilizadas:

- Java: El backend del proyecto está desarrollado en Java, utilizando servlets y JDBC para la conexión con la base de datos. Se utilizó la arquitectura MVC (Modelo-Vista-Controlador) para estructurar el código y separar las responsabilidades.
- Angular: La interfaz de usuario (frontend) está desarrollada en Angular, un framework de JavaScript que permite construir aplicaciones web dinámicas y eficientes. Angular se utilizó para gestionar el flujo de datos y las interacciones entre el usuario y el backend.
- SQL: El sistema utiliza SQL para interactuar con la base de datos MySQL.
 Las consultas SQL se emplean para recuperar y modificar datos en las tablas de la base de datos, como la información de los anuncios, usuarios, y balances de los anunciantes.
- MySQL: Se utiliza MySQL como sistema de gestión de bases de datos para almacenar la información del sistema. Las tablas principales incluyen Anunciante, Anuncio, y Cartera.
- NetBeans y Visual Studio Code: El desarrollo del backend se llevó a cabo utilizando NetBeans, mientras que el frontend se desarrolló utilizando Visual Studio Code. Ambos entornos proporcionaron un entorno robusto para escribir, depurar y probar el código.
- MySQL Workbench: Para la gestión de la base de datos, se utilizó
 MySQL Workbench. Esta herramienta proporciona una interfaz gráfica para diseñar y gestionar las bases de datos, ejecutar consultas SQL, y administrar los datos almacenados.

3. Estructura del Proyecto:

El proyecto se divide en dos secciones principales: **Backend** y **Frontend**.

Backend (Java):

- La lógica del backend está implementada utilizando serviets y JDBC.
- El backend se encarga de procesar las solicitudes del frontend, interactuar con la base de datos y enviar respuestas al cliente.
- Ejemplos de funcionalidades del backend:
 - Gestión de anuncios: Creación, visualización y eliminación de anuncios.
 - Verificación de la cartera del anunciante.

Cálculo y validación del pago según la duración del anuncio.

• Frontend (Angular):

- La interfaz de usuario es una aplicación Angular que interactúa con el backend a través de AJAX y HTTP.
- El frontend permite a los usuarios:
 - Crear y visualizar anuncios.
 - Subir imágenes y videos.
 - Consultar los anuncios disponibles.
 - Consultar el saldo de su cartera.

4. Flujo de Trabajo:

1. Creación de Anuncios:

- El usuario (anunciante) ingresa al sistema y crea un anuncio seleccionando el tipo de contenido (texto, imagen, video).
- La solicitud se envía al servidor, donde se valida el tipo de anuncio y el saldo del anunciante.
- Si el saldo es suficiente, el anuncio se guarda en la base de datos y se muestra en la interfaz.

2. Visualización de Anuncios:

- El usuario puede visualizar los anuncios creados en el sistema.
- Dependiendo del tipo de anuncio (texto, imagen o video), el sistema muestra el contenido adecuado en el frontend.

3. Verificación de Saldo:

 Al crear un anuncio, se verifica que el anunciante tenga suficiente saldo para cubrir el costo del anuncio. Si no tiene saldo suficiente, se muestra un mensaje de error.

5. Arquitectura:

Modelo-Vista-Controlador (MVC):

- Modelo: Contiene la lógica de negocio y acceso a datos (por ejemplo, las clases que interactúan con la base de datos).
- Vista: La interfaz de usuario que permite la interacción con el usuario (desarrollada en Angular).
- Controlador: Los servlets en el backend que manejan las solicitudes y respuestas entre la vista y el modelo.

6. Base de Datos:

El sistema utiliza una base de datos MySQL con las siguientes tablas principales:

 Anunciante: Almacena la información del anunciante, incluyendo el saldo de la cartera.

- Columnas: idAnunciante, nombre_usuario, cartera, nombre_anunciante, etc.
- **Anuncio**: Almacena los anuncios creados por los anunciantes.
 - Columnas: idAnuncio, texto, tipo, imagen, video, fecha_inicio, fecha_fin, estado, etc.

7. Implementación del Backend:

El backend se implementó en Java utilizando servlets para procesar las solicitudes HTTP y realizar las operaciones con la base de datos. Aquí se explica un ejemplo de cómo se maneja la creación de anuncios:

```
@WebServlet(name = "CrearAnuncio", urlPatterns = {"/CrearAnuncio"})
public class CrearAnuncioServlet extends HttpServlet {
  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
     // Obtener los datos del formulario
     String nombreAnunciante = request.getParameter("nombreAnunciante");
     String tipo = request.getParameter("tipo");
     String texto = request.getParameter("texto");
     int diasDuracion = Integer.parseInt(request.getParameter("diasDuracion"));
     // Validación del saldo
     int cartera = obtenerCartera(nombreAnunciante);
     int pago = calcularPago(diasDuracion);
     if (pago > cartera) {
       response.setStatus(HttpServletResponse.SC_BAD_REQUEST);
       response.getWriter().write("No tienes fondos suficientes");
       return;
    }
    // Crear el anuncio
     Anuncio anuncio = new Anuncio(tipo, texto, pago, nombreAnunciante);
     guardarAnuncio(anuncio);
    // Responder con éxito
     response.getWriter().write("Anuncio creado con éxito");
  }
  private int obtenerCartera(String nombreUsuario) {
    // Lógica para obtener el saldo de la cartera del anunciante
     return 500; // Ejemplo de saldo
  }
  private int calcularPago(int diasDuracion) {
```

```
return diasDuracion * 10;
}

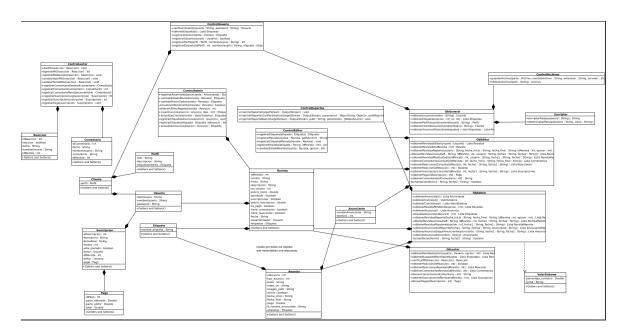
private void guardarAnuncio(Anuncio anuncio) {
    // Lógica para guardar el anuncio en la base de datos
}
}
```

8. Implementación del Frontend:

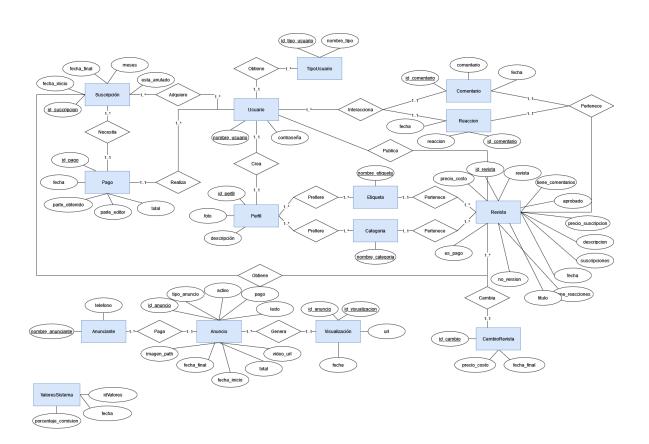
En el frontend, se utiliza Angular para gestionar la interacción con el usuario. Un ejemplo de cómo se puede mostrar un anuncio es el siguiente:

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
@Component({
 selector: 'app-anuncios',
 templateUrl: './anuncios.component.html',
 styleUrls: ['./anuncios.component.css']
})
export class AnunciosComponent implements OnInit {
 anuncios: any[] = [];
 constructor(private http: HttpClient) {}
 ngOnInit(): void {
  this.cargarAnuncios();
 }
 cargarAnuncios(): void {
  this.http.get('/Backend/resources/anuncios')
    .subscribe((data: any) => {
     this.anuncios = data;
   });
 }
}
```

9. Diagrama de clases:



10. Diagrama E/R



11. Diagrama de tablas:

