2025

Documentación Proyecto 1ºDAW

JOSE ENRIQUE ÁGUILA BLANCO

Contenido

1. Introducción

1.1 Objetivo del Proyecto

ECOLIMBO es una plataforma web desarrollada con el propósito de facilitar la organización, promoción y gestión de eventos ambientales. Su diseño permite que tanto usuarios como organizadores interactúen de manera eficiente, accediendo a información sobre eventos próximos, inscribiéndose en actividades y gestionando la planificación de nuevas iniciativas ecológicas.

1.2 Justificación

En la actualidad, la concienciación sobre la sostenibilidad y la protección del medioambiente es una prioridad. ECOLIMBO busca convertirse en un espacio centralizado donde los ciudadanos puedan participar activamente en actividades ecológicas, fortaleciendo el impacto de estas iniciativas mediante una mejor organización y difusión.

1.3 Tecnologías Utilizadas

El desarrollo de ECOLIMBO se basa en un conjunto de tecnologías modernas que garantizan su eficiencia y escalabilidad:

- **Frontend:** HTML y CSS para la interfaz de usuario.
- **Backend:** Kotlin para la lógica del servidor y la gestión de eventos.
- **Base de datos:** SQL para almacenar información sobre usuarios y eventos.
- **Servidor local:** XAMPP para pruebas y despliegue en entornos de desarrollo.
- **Control de versiones:** GitHub para la gestión del código fuente.

2. Arquitectura del Sistema

El sistema se compone de tres capas principales:

1. **Capa de Presentación (Frontend):** Contiene las interfaces de usuario en HTML y CSS.
2. **Capa de Lógica (Backend):** Implementada en Kotlin, maneja la lógica de negocio.
3. **Capa de Datos (Base de Datos):** Responsable del almacenamiento y recuperación de información.

Flujo de trabajo:

1. Un usuario u organizador accede al portal y visualiza los eventos disponibles.
2. Se registra o inicia sesión.
3. Puede inscribirse en eventos o, si es organizador, crear nuevos.
4. La información es almacenada en la base de datos y reflejada en la interfaz.

3. Estructura del Proyecto

3.1 HTML

Los archivos HTML proporcionan la estructura de la aplicación:

- **Inicio_org.html:** Página de inicio para organizadores con opciones de filtrado de eventos.
- **inicio_sesion.html:** Página de autenticación.
- **Registro.html:** Formulario de registro para usuarios y organizadores.
- **Inicio.html:** Página principal de eventos para usuarios generales.

3.2 CSS

El archivo **styles.css** define los estilos y la apariencia del sitio web, incluyendo:

- Temas diferenciados para usuarios y organizadores.
- Diseño responsivo adaptable a distintos dispositivos.
- Estilización de botones, formularios y listados de eventos.

3.3 Base de Datos y Modelado

3.3.1 Esquema Entidad-Relación

La base de datos se estructura en torno a las siguientes entidades:

- **USUARIO:** Información de los usuarios registrados.
- **EVENTO:** Datos de los eventos organizados.
- **ORGANIZADOR:** Usuarios con permisos para gestionar eventos.
- **CATEGORIA:** Clasificación de eventos.

3.3.2 Relaciones Claves

Las relaciones dentro del modelo de datos incluyen:

- **INSCRIPCIÓN:** Un usuario se inscribe en un evento.
- **ASISTENCIA:** Registro de participación en eventos.
- **ORGANIZACIÓN:** Asociación entre organizadores y eventos.
- **CAT_EVENTO:** Relación entre eventos y categorías.

4. Programación

El backend está desarrollado en **Kotlin**, utilizando una estructura modular y orientada a objetos.

4.1 Clases Principales

- **Categoría:** Enum con los tipos de eventos disponibles.
- **Constante:** Definición de valores constantes para fechas y contraseñas.
- **Evento:** Clase base para modelar eventos.
- **Factoría:** Genera instancias de usuarios, organizadores y eventos.
- **Main:** Punto de entrada del programa, ejecutando la lógica de negocio.
- **Online y Presencial:** Subclases de Evento para diferenciar eventos virtuales y presenciales.
- **Organizador:** Representa a los organizadores de eventos.
- **Usuario:** Implementa la lógica de inscripción en eventos.

4.2 Flujo del Programa

1. Se crean usuarios y organizadores mediante **Factoría**.
2. Se generan eventos aleatorios (presenciales y online).
3. Se listan los eventos disponibles.
4. Los usuarios pueden inscribirse en eventos.
5. Se imprime la información en consola para depuración.

5. Instalación y Configuración

5.1 Instalación de Windows 10 en Máquina Virtual

1. Descargar la imagen ISO de Windows 10.
2. Configurar una máquina virtual en Oracle VirtualBox.
3. Asignar almacenamiento y memoria.
4. Realizar la instalación estándar del sistema operativo.

5.2 Instalación y Configuración de XAMPP

1. Descargar XAMPP desde su página oficial.
2. Instalar y configurar Apache y MySQL.
3. Verificar acceso mediante localhost.

6. Funcionalidades Claves

6.1 Autenticación de Usuarios

- Registro de usuarios y organizadores.
- Inicio de sesión seguro.

6.2 Gestión de Eventos

- Visualización y filtrado de eventos.
- Inscripción en eventos.
- Creación y administración de eventos para organizadores.

7. Buenas Prácticas y Posibles Mejoras

Para optimizar el rendimiento y escalabilidad del sistema, se sugieren las siguientes mejoras:

- **Implementación de autenticación segura:** Uso de JWT o OAuth para la gestión de sesiones.
- **Persistencia de datos optimizada:** Integración con bases de datos relacionales como PostgreSQL.
- **Interfaz más dinámica:** Uso de frameworks como React o Vue.js.
- **Automatización del despliegue:** Uso de contenedores Docker y CI/CD.