



PORTAL WEB PARA EVENTOS SOSTENIBLES

SERGIO ZARCERO PEREZ



28 DE FEBRERO DE 2025

PROYECTO PRACTICAS 2025

INDRA

Documentación General del Proyecto

1. INTRODUCCION

NOMBRE DEL PROYECTO:

Portal para la Gestión de Eventos Sostenibles

DESCRIPCIÓN GENERAL:

El proyecto consiste en el desarrollo de un portal web que permite la gestión y organización de eventos relacionados con la sostenibilidad, como conferencias, talleres y actividades ecológicas. la plataforma está diseñada para que los usuarios puedan registrarse, inscribirse en eventos y consultar información detallada sobre cada actividad. además, los organizadores podrán gestionar sus eventos, modificarlos y clasificar sus actividades acorde con las categorías predefinidas.

el sistema se implementará con una combinación de kotlin para la lógica del backend, html y css para la interfaz de usuario, y una base de datos sql para almacenar la información de los eventos, usuarios y sus inscripciones.

OBJETIVO DEL PROYECTO:

El principal objetivo es desarrollar una aplicación funcional y automatizada que facilite la gestión de eventos sostenibles, permitiendo a los organizadores crear y modificar eventos, y a los usuarios inscribirse de manera sencilla. Además, se busca aplicar los conocimientos adquiridos en desarrollo web, bases de datos, programación, despliegue web y control de versiones en un contexto realista.

TECNOLOGÍAS UTILIZADAS:

Para el desarrollo del proyecto vamos a utilizar las siguientes tecnologías.

Backend: Kotlin

Frontend: HTML y CSS

Base de Datos: SQL (gestión a través de un modelo entidad-relación)

Servidor Web: XAMPP para el despliegue en un entorno Windows 10 virtualizado

Control de Versiones: Git & GitHub para gestionar el código y la documentación del proyecto.

1.1 ALCANCE DEL PROYECTO

EL SISTEMA DEBERÁ PERMITIR:

Gestión de eventos: Creación, modificación y cancelación de eventos por parte de los organizadores.

Registro de usuarios: Los usuarios podrán registrarse e iniciar sesión en la plataforma.

Inscripción a eventos: Los usuarios podrán apuntarse y cancelar su participación en eventos.

Clasificación de eventos: Los eventos estarán distinguidos en diferentes categorías para facilitar su búsqueda.

Control de accesos y permisos: Se garantizará que cada usuario solo pueda realizar las acciones que le correspondan.

Este web será desplegada en un servidor local con XAMPP dentro de una máquina virtual con Windows 10, documentaremos el proceso con diferentes capturas de pantalla de los pasos que sigamos en este proceso.

LIMITACIONES DEL PROYECTO:

- La primera versión del portal será un prototipo funcional sin integración con pasarelas de pago.
- Los eventos solo podrán ser gestionados por organizadores registrados.
- El portal no incluirá funcionalidades avanzadas como recordatorios automáticos o integración con redes sociales en esta etapa.

1.2 ARQUITECTURA DEL PROYECTO.

El Portal para la Gestión de Eventos Sostenibles está diseñado siguiendo una arquitectura modular que permite la separación de responsabilidades entre el *frontend*, el *backend* y la *base de datos*. El sistema sigue un enfoque estructurado que facilita la escalabilidad, el mantenimiento y la implementación de nuevas funcionalidades en el futuro.

ESTRUCTURA GENERAL

El sistema se divide en tres capas principales:

1. CAPA DE PRESENTACIÓN (FRONTEND)

- Se desarrolla con HTML y CSS.
- Contiene la interfaz gráfica con las páginas web donde los usuarios pueden interactuar.
- Presenta la información de los eventos, permite el registro de usuarios y la inscripción en eventos.

2. CAPA LÓGICA (BACKEND)

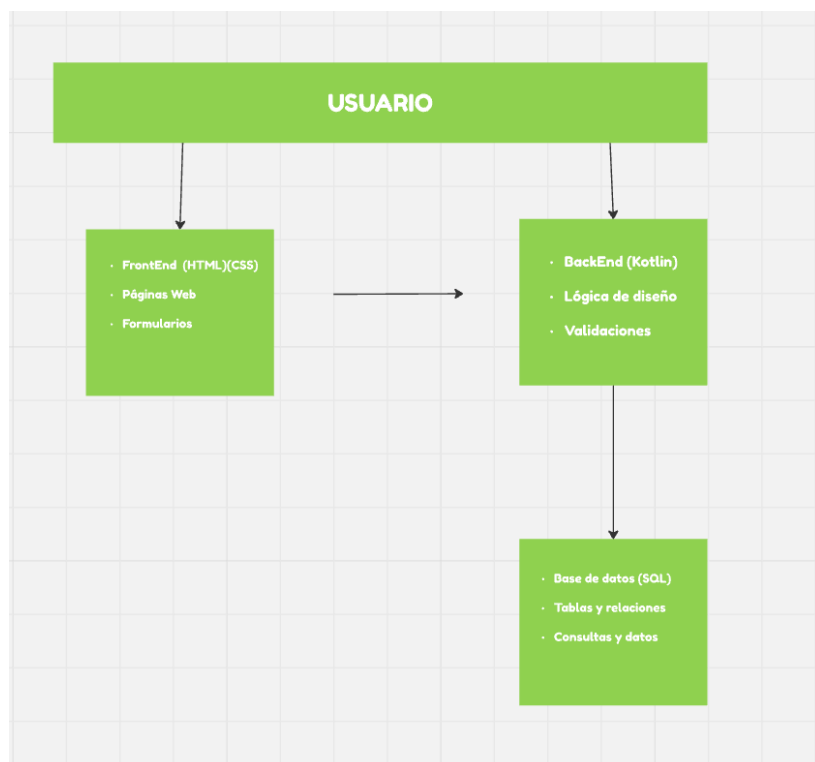
- Desarrollada en Kotlin.
- Se encarga de la lógica del negocio, validaciones y procesamiento de datos.
- Gestiona las solicitudes del frontend y realiza operaciones sobre la base de datos.

3. CAPA DE DATOS (BASE DE DATOS)

- Implementada en SQL.
- Contiene las tablas de eventos, usuarios, organizadores e inscripciones.
- Se relaciona con el backend a través de consultas SQL.
- Asegura la integridad y consistencia de los datos almacenados.

DIAGRAMA DE ARQUITECTURA

El siguiente diagrama representa la interacción entre los componentes del sistema:



FLUJO DE INTERACCIÓN

- **El usuario accede al portal** a través del navegador y visualiza la página principal desarrollada en HTML y CSS.
- **Accede o se registra** en el sistema proporcionando sus credenciales.
- **Consulta los eventos disponibles** desde una lista generada dinámicamente.
- **Selecciona un evento** para ver más detalles y decidir si se inscribe.
- **El backend procesa la inscripción**, registrando la relación entre el usuario y el evento en la base de datos.
- **Los organizadores pueden gestionar los eventos**, modificando su información o cancelándolos si es necesario.
- **El sistema almacena toda la información en la base de datos** y la actualiza cuando se realizan cambios.

2. PLANIFICACIÓN DEL DESARROLLO Y GESTIÓN DEL PROYECTO

En esta sección se describe cómo se organizó y gestionó el desarrollo del Portal para la Gestión de Eventos Sostenibles, desde la planificación inicial hasta la administración del código en GitHub.

2.1 METODOLOGÍA DE TRABAJO

El desarrollo del proyecto se llevó a cabo utilizando una metodología ágil basada en Sprints, permitiendo un flujo de trabajo iterativo y adaptable a nuevas necesidades.

METODOLOGÍA APLICADA

- **Gestión de tareas:** Se utilizó **GitHub Projects** con un tablero, donde las tareas se clasificaron en:
 - **Backlog:** Pendientes de iniciar.
 - **In Progress:** En proceso de desarrollo.
 - **Done:** Completadas.

- **Ciclos de desarrollo:** Se definieron Sprints semanales, asignando tareas a cada iteración.
- **División del trabajo:** Se estructuró el desarrollo en tres áreas principales:
 - **Frontend (HTML/CSS).**
 - **Backend (Kotlin).**
 - **Base de Datos (SQL).**

2.2 CREACIÓN DEL REPOSITORIO Y CONTROL DE VERSIONES

Para la gestión del código se creó **un** repositorio en GitHub, donde se almacenan todos los archivos del proyecto, permitiendo el seguimiento de cambios y colaboración eficiente.

PASOS SEGUIDOS PARA LA CREACIÓN DEL REPOSITORIO EN GITHUB:

- Se accedió a **GitHub** y se creó un nuevo repositorio llamado "**Proyecto prácticas 1ºDAW**".
- Se clonó el repositorio en local con: GitHub Desktop
- Se vinculó el repositorio local con: GitHub Desktop

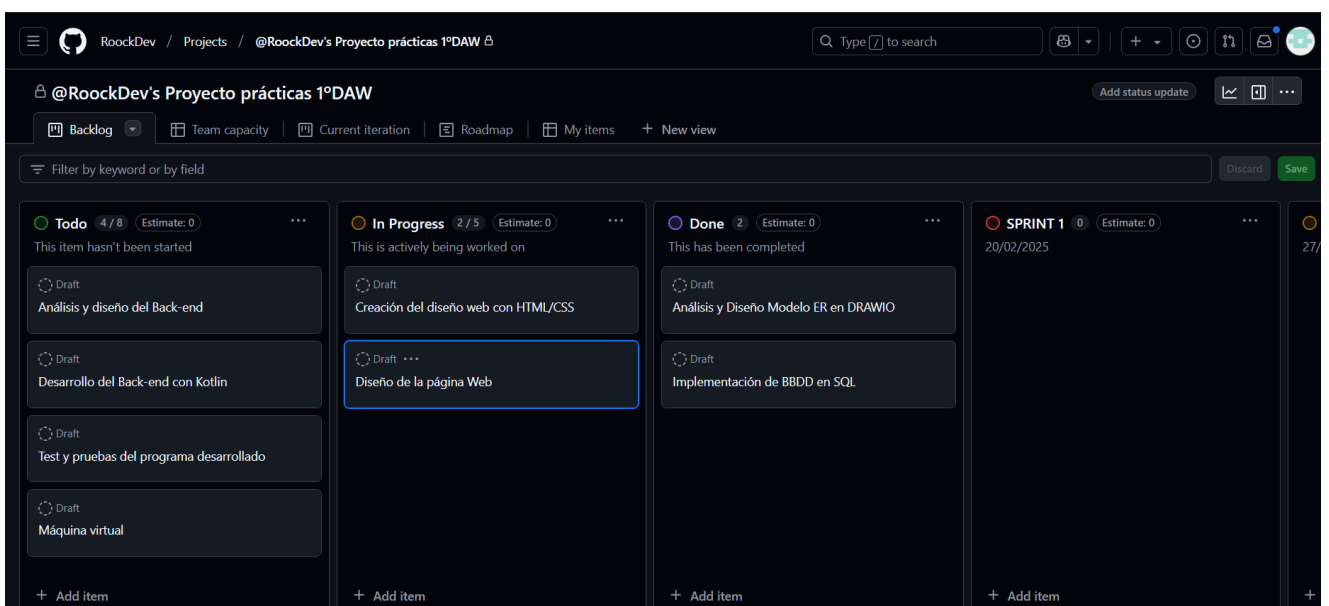
2.3 ORGANIZACIÓN DEL PROYECTO EN GITHUB PROJECTS

Se utilizó **GitHub Projects** para gestionar el desarrollo, estructurando las tareas en diferentes estados:

ESTADOS DEL TABLERO :

- **Backlog (Pendiente):**
 - Análisis y diseño del Backend.
 - Desarrollo del Backend con Kotlin.
 - Análisis y Diseño del Modelo ER en Draw.io.
 - Implementación de la Base de Datos en SQL.
 - Creación del diseño web con HTML/CSS.
 - Diseño de la página web.
 - Test y pruebas del programa.
 - Configuración de máquina virtual.
- **In Progress (En Desarrollo):** Aquí asignaremos las tareas que estemos realizando en el momento.
- **Done (Completado):** En este apartado asignaremos las tareas que ya hemos completado, llevando así un orden.
- **Sprints:** Aquí asignaremos las tareas que correspondan y deban ser entregadas en esa fecha para llevar un orden de los días en los que se entregaron las tareas correspondientes.

(Se incluye una captura de pantalla del tablero de GitHub Projects para evidenciar la organización del proyecto).



3. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

El sistema del **Portal para la Gestión de Eventos Sostenibles** requiere una base de datos robusta que almacene información sobre eventos, usuarios, organizadores e inscripciones. En esta sección se detalla el diseño, la implementación y la gestión de la base de datos utilizada en el proyecto.

Para desarrollar el esquema Entidad relación hemos observado y analizado las necesidades del cliente para el portal de la gestión de eventos sostenibles.

El cliente quiere un sistema para gestionar eventos relacionados con la sostenibilidad, tales como conferencias talleres etc. El cliente quiere una solución que permita gestionar de manera automática la inscripción de los usuarios, el conteo de usuarios que se inscriben y asisten, la organización de eventos, Clasificación de eventos y la asignación de organizadores de los eventos.

Modelado de Base de Datos

3.1 ENTIDADES

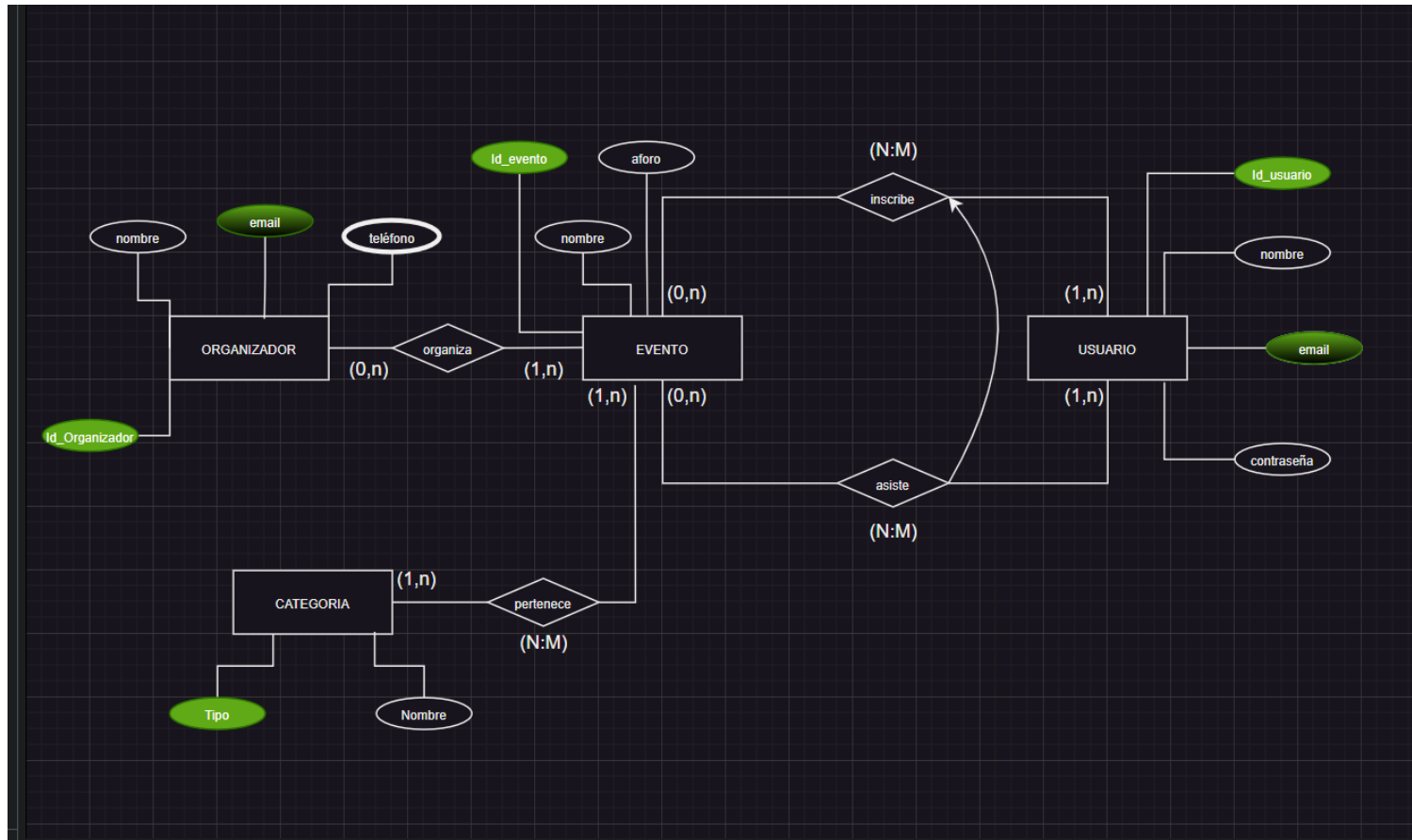
HEMOS IDENTIFICADO LAS SIGUIENTES ENTIDADES:

- **Usuario:** representa a los usuarios que se registran en el sistema.
- **Organizador:** representa a las personas o empresas encargadas de organizar los eventos.
- **Evento:** representa cada evento registrado en el sistema.
- **Categoría:** representa la categoría a la que pertenece un evento.

RELACIÓN ENTRE ENTIDADES.

- ***Usuario-Inscribe-Evento:*** un usuario puede inscribirse a varios eventos y un evento puede tener varios inscritos. (N:M)
- ***Usuario-asiste-Evento:*** un usuario puede asistir a varios eventos y un evento puede tener varios asistentes. (N:M)
- ***Organizador-organiza-Evento:*** Un organizador Un organizador puede gestionar varios eventos y cada evento tiene al menos un organizador y puede ser organizador por varios organizadores. (N:M)
- ***Categoría-pertenece-Evento:*** Un evento puede tener varias categorías y una categoría puede pertenecer a muchos eventos. (N:M)

ESQUEMA ENTIDAD RELACION



3.2 IMPLEMENTACIÓN EN SQL

La base de datos fue implementada en Oracle mediante un script SQL que crea las tablas y define sus relaciones. A continuación, se muestra el código SQL para la creación de las principales tablas:

```
CREATE TABLE USUARIO(  
ID_USUARIO NUMBER(5) GENERATED ALWAYS AS IDENTITY  
PRIMARY KEY,  
NOMBRE VARCHAR2(30) NOT NULL,  
EMAIL VARCHAR2(30) UNIQUE,  
CONTRASEÑA VARCHAR(30) NOT NULL  
);  
  
CREATE TABLE EVENTO(  
ID_EVENTO NUMBER(7) GENERATED ALWAYS AS IDENTITY  
PRIMARY KEY,  
NOMBRE VARCHAR2(30) NOT NULL,  
AFORO NUMBER(5) NOT NULL  
);  
  
CREATE TABLE ORGANIZADOR(  
ID_ORGANIZADOR NUMBER(7) GENERATED ALWAYS AS IDENTITY  
PRIMARY KEY,  
NOMBRE VARCHAR2(30) NOT NULL,  
EMAIL VARCHAR2(30) UNIQUE,  
TELEFONO NUMBER(13) NOT NULL  
);  
  
CREATE TABLE CATEGORIA(  
TIPO VARCHAR2(30) PRIMARY KEY,  
NOMBRE VARCHAR2(30) NOT NULL  
);  
  
CREATE TABLE INSCRIPCION(  
ID_USUARIO NUMBER(5),  
ID_EVENTO NUMBER(7),  
PRIMARY KEY(ID_USUARIO, ID_EVENTO),  
CONSTRAINT FK_INS_USU FOREIGN KEY  
(ID_USUARIO) REFERENCES USUARIO(ID_USUARIO),  
CONSTRAINT FK_INS_EVENT FOREIGN KEY
```

```

(ID_EVENTO) REFERENCES EVENTO(ID_EVENTO)
);

CREATE TABLE ASISTENCIA(
ID_USUARIO NUMBER(5),
ID_EVENTO NUMBER(7),
PRIMARY KEY(ID_USUARIO, ID_EVENTO),
ASISTE VARCHAR2(2) CHECK (ASISTE IN ('SI', 'NO')) NOT NULL,
CONSTRAINT FK_ASIS_USU FOREIGN KEY
(ID_USUARIO) REFERENCES USUARIO(ID_USUARIO),
CONSTRAINT FK_ASIS_EVENT FOREIGN KEY
(ID_EVENTO) REFERENCES EVENTO(ID_EVENTO)
);

CREATE TABLE ORGANIZACION(
ID_EVENTO NUMBER(5),
ID_ORGANIZADOR NUMBER(7),
PRIMARY KEY(ID_EVENTO, ID_ORGANIZADOR),
CONSTRAINT FK_ORGANI FOREIGN KEY
(ID_ORGANIZADOR) REFERENCES ORGANIZADOR(ID_ORGANIZADOR),
CONSTRAINT FK_ORGANI_EVENT FOREIGN KEY
(ID_EVENTO) REFERENCES EVENTO(ID_EVENTO)
);

CREATE TABLE CAT_EVENTO(
ID_EVENTO NUMBER(7),
TIPO VARCHAR2(30),
PRIMARY KEY(ID_EVENTO, TIPO),
CONSTRAINT FK_TIPO FOREIGN KEY
(TIPO) REFERENCES CATEGORIA(TIPO),
CONSTRAINT FK_CAT_EVENT FOREIGN KEY
(ID_EVENTO) REFERENCES EVENTO(ID_EVENTO)
);

```

3.3 CONCLUSIÓN

El diseño de la base de datos permite un almacenamiento eficiente de la información del sistema, asegurando que los eventos, usuarios y organizadores se gestionen de manera estructurada. Su integración con el backend en Kotlin garantiza una comunicación fluida y una experiencia de usuario optimizada.

4. DESARROLLO DEL FRONTEND Y GESTIÓN DE EVENTOS

El sistema del Portal para la Gestión de Eventos Sostenibles incluye una interfaz web intuitiva y accesible, desarrollada con HTML y CSS, permitiendo a los usuarios explorar, inscribirse y crear eventos de manera sencilla.

A continuación, se detalla la estructura y funcionalidad de cada página web implementada.

4.1 TECNOLOGÍAS UTILIZADAS

El frontend se desarrolló con las siguientes tecnologías:

- **HTML5:** Estructuración del contenido y elementos de la interfaz.
- **CSS3:** Estilos y diseño.
- **Flexbox :** Para organizar los elementos de la página de forma eficiente.
- **Efectos CSS:** Se aplicaron efectos de hover y animaciones para mejorar la experiencia del usuario.

4.2 PÁGINAS IMPLEMENTADAS

El sistema cuenta con un total de 15 páginas que conforman el portal web:

4.2.1 PÁGINA DE INICIO (INDEX.HTML)

Descripción: Página principal del portal donde los usuarios pueden navegar entre las distintas secciones y explorar eventos sostenibles.

Características:

- Presentación general del proyecto.
- Botón para crear un evento.
- Botón para explorar los eventos disponibles.
- Información sobre el impacto ambiental y el uso de IA en eventos.

4.2.2 PÁGINA DE CREACIÓN DE EVENTOS (CREA-TU-EVENTO.HTML)

Descripción: Permite a los organizadores registrar un nuevo evento, rellenando un formulario.

Características:

- Formulario con campos para ingresar:
 - Nombre del organizador.
 - Correo electrónico.
 - Nombre del evento.
 - Categoría del evento (Ciberseguridad, IA, Realidad Virtual, etc.).
 - Público objetivo (Empresas, Estudiantes, Comunidad).
 - Contenidos adicionales del evento (Charlas, Talleres, etc.).
- Botón para enviar la solicitud de registro.

4.2.3 PÁGINA DE LISTA DE EVENTOS (LISTA-EVENTOS.HTML)

Descripción: Muestra un catálogo con los eventos disponibles, permitiendo a los usuarios conocer detalles y registrarse.

Características:

- Sección con tarjetas de eventos que muestran:
 - Nombre del evento.
 - Descripción breve.
 - Botón para acceder a la página detallada del evento.
- Cada tarjeta tiene un efecto de hover para resaltar la interacción.
- Botón para crear un evento desde esta página.

4.2.4 PÁGINAS DE EVENTOS DETALLADOS (EVENTO1.HTML, EVENTO2.HTML, ETC.)

Descripción: Cada evento cuenta con su propia página con detalles completos.

Características:

- Información detallada del evento:
 - Nombre, fecha y ubicación.
 - Descripción extendida.
- Botón para inscribirse en el evento.
- Botón para volver a la lista de eventos.
- Fondo personalizado con una imagen representativa del evento.

Ejemplo de eventos creados:

1. **"AI Green Cities"** - Uso de IA en ciudades sostenibles.
2. **"EcoBot Challenge"** - Competencia de robots recicladores.
3. **"Smart Farming"** - IA aplicada a la agricultura.
4. **"AI para la Sostenibilidad Energética"** - IA sostenible.

4.2.5 PÁGINA DE INSCRIPCIÓN (INSCRIPCION.HTML)

Descripción: Formulario donde los usuarios pueden inscribirse en los eventos.

Características:

- Campos requeridos:
 - Nombre y apellidos.
 - Correo electrónico.
 - Número de teléfono.
 - Selección del evento al que desean inscribirse.
- Opción de añadir complementos, como:
 - Camiseta del evento.
 - Talleres adicionales.
 - Certificado de asistencia.

4.2.6 PÁGINA DE INICIO DE SESIÓN (INICIO-SESION.HTML)

Descripción: Permite a los usuarios registrados acceder a su cuenta.

Características:

- Campos para nombre de usuario y contraseña.
- Botón para recuperar la contraseña.
- Enlace para registrarse si no tienen cuenta.

4.2.7 PÁGINA DE REGISTRO (REGISTRO.HTML)

Descripción: Permite a nuevos usuarios crear una cuenta.

Características:

- Campos requeridos:
 - Nombre completo.
 - Correo electrónico.
 - Contraseña segura.
- Opción para registrarse con un solo clic.

4.3 DISEÑO Y ESTILOS

El diseño del portal se optimizó para ofrecer una experiencia de usuario fluida y atractiva, aplicando los siguientes principios:

Uso de colores sostenibles

- Fondo en tonos verdes y blancos para representar la sostenibilidad.
- Botones con colores llamativos para mejorar la accesibilidad.

Animaciones y efectos

- Se implementaron efectos de hover y escalado en botones y tarjetas.
- Opacidad en imágenes de fondo para resaltar el contenido.

PÁGINA DE INICIO (INDEX.HTML)

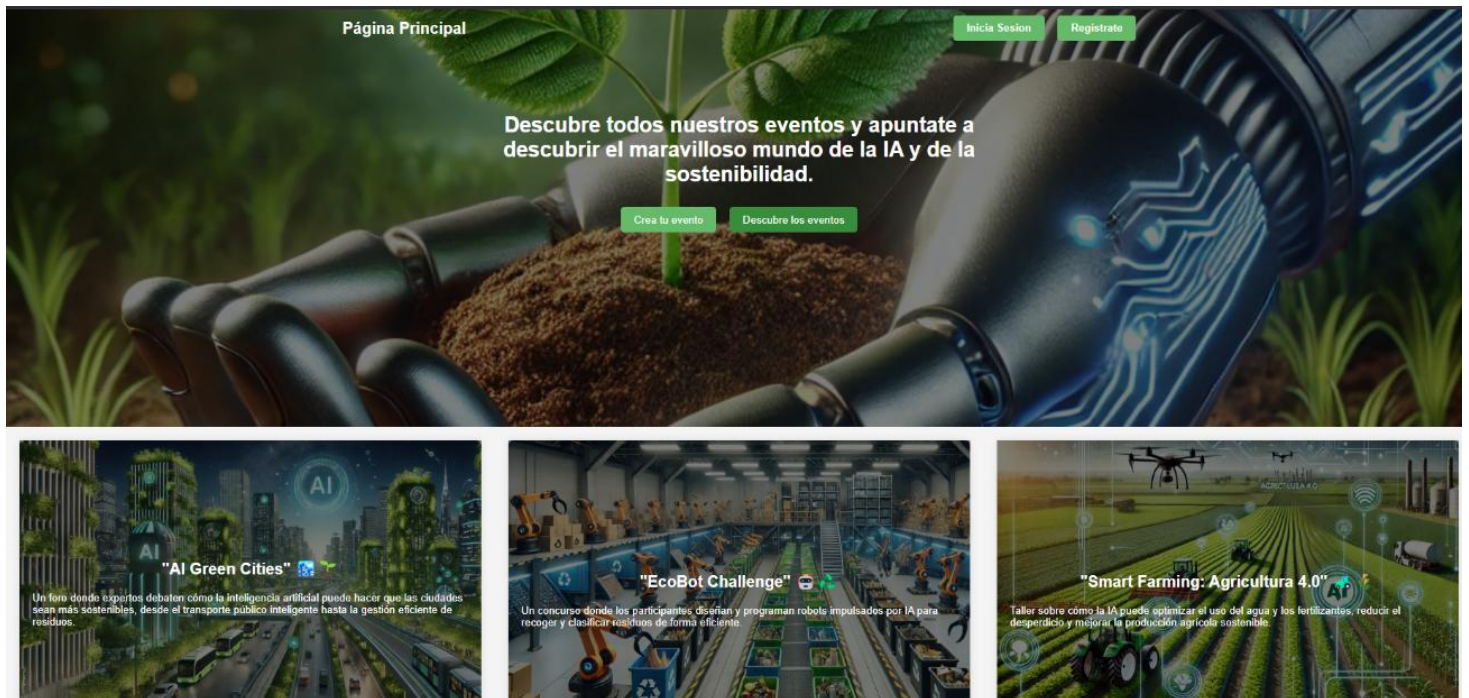


PÁGINA DE CREACIÓN DE EVENTOS (CREA-TU-EVENTO.HTML)

The event creation page features a dark green background with a robotic hand holding a small green plant. The main heading is "Diseña el futuro: Crea tu evento sostenible". Below it is a subheading: "Rellena el siguiente formulario con la información de tu evento". The form contains the following fields:

- Nombre del organizador o de la empresa organizadora
Introduce el Nombre
- Email
Introduce tu email
- Nombre del evento
Nombre del evento
- ¿Que tipo de evento quieres crear?
Selecciona una opción
- ¿Colaborarías con otro organizador?
☐ Si
☐ Quizás
☒ No estoy seguro

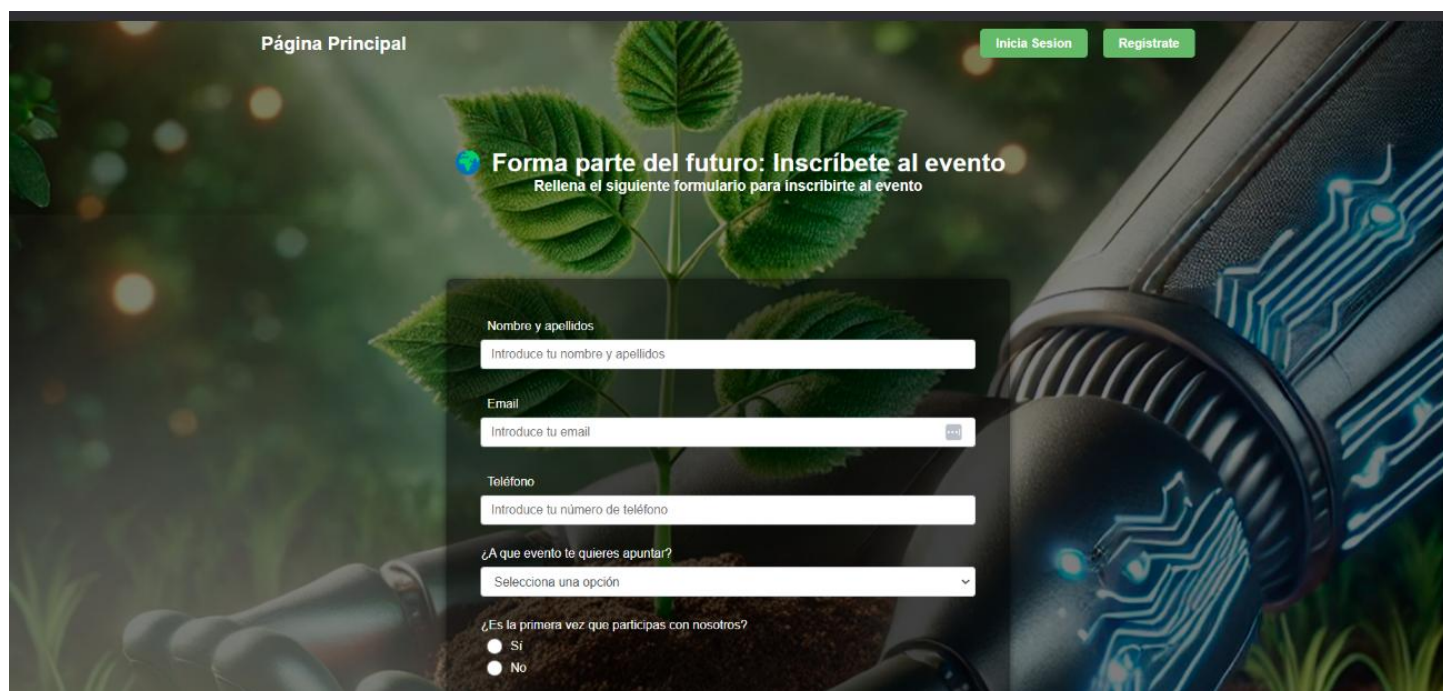
PÁGINA DE LISTA DE EVENTOS (LISTA-EVENTOS.HTML)



PÁGINAS DE EVENTOS DETALLADOS (EVENTO1.HTML, EVENTO2.HTML, ETC.)




PÁGINA DE INSCRIPCIÓN (INSCRIPCION.HTML)



Página Principal

Inicia Sesión Registrarse

 **Forma parte del futuro: Inscríbete al evento**
Rellena el siguiente formulario para inscribirte al evento

Nombre y apellidos
Introduce tu nombre y apellidos

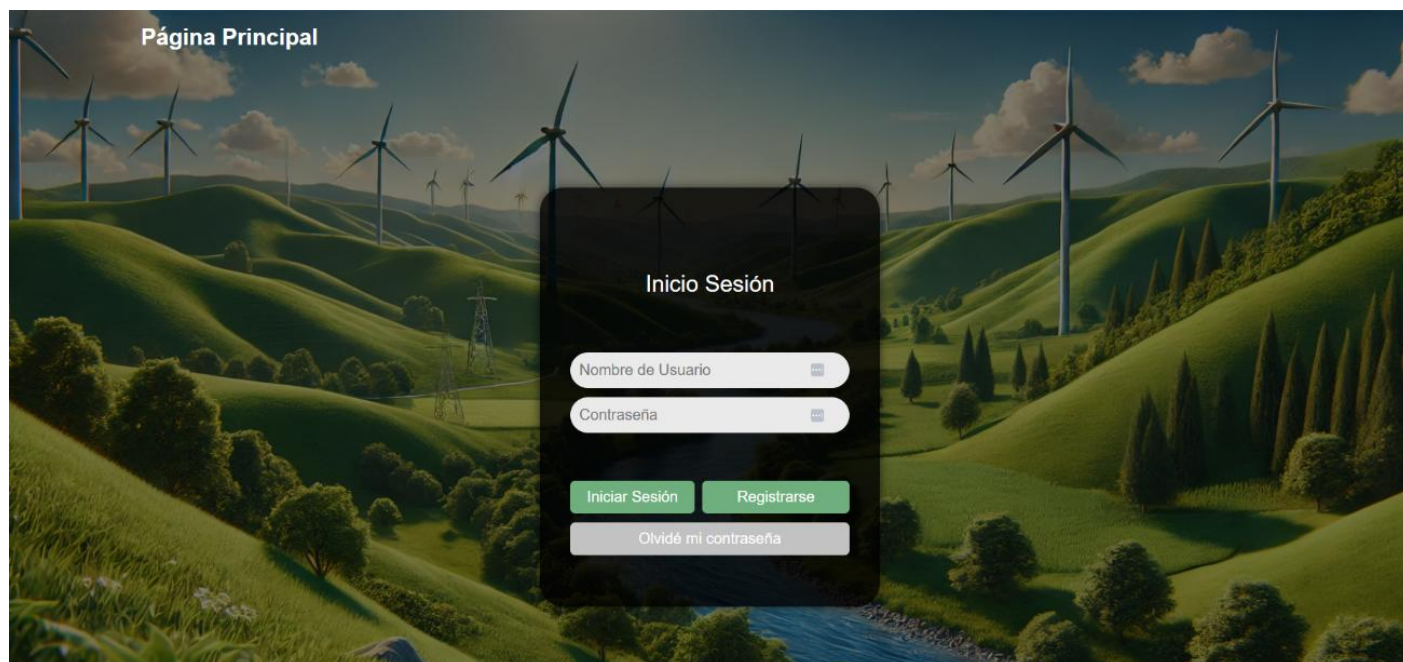
Email
Introduce tu email

Teléfono
Introduce tu número de teléfono

¿A que evento te quieres apuntar?
Selecciona una opción

¿Es la primera vez que participas con nosotros?
☐ Sí
☐ No

PÁGINA DE INICIO DE SESIÓN (INICIO-SESION.HTML)



Página Principal

Inicio Sesión

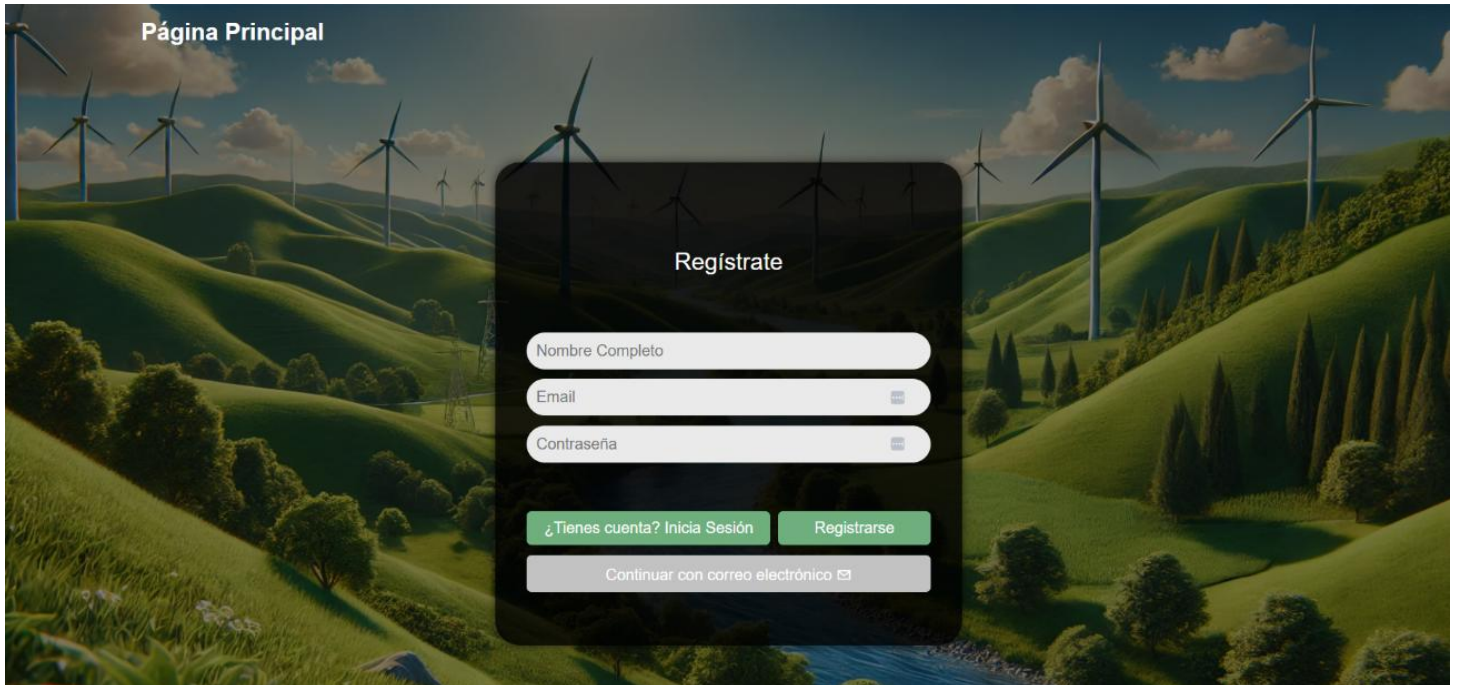
Nombre de Usuario

Contraseña

Iniciar Sesión Registrarse

Olvidé mi contraseña

PÁGINA DE REGISTRO (REGISTRO.HTML)



4.5 CONCLUSIÓN

El desarrollo del frontend garantiza que los usuarios puedan navegar intuitivamente por el portal, registrarse, explorar eventos e inscribirse de manera sencilla. Con una interfaz moderna, atractiva y accesible, la plataforma está preparada para su futura integración con el backend.

5. DESARROLLO DEL BACKEND Y SIMULACIÓN DE EVENTOS

El backend del Portal para la Gestión de Eventos Sostenibles ha sido desarrollado en Kotlin, implementando la lógica de negocio necesaria para gestionar usuarios, organizadores y eventos.

Debido a que aún no hemos implementado una base de datos real, he desarrollado un sistema de simulación que genera datos aleatorios para los eventos, usuarios y organizadores, permitiendo probar el funcionamiento del sistema sin necesidad de una conexión a una base de datos.

5.1 OBJETIVOS DEL BACKEND

El backend tiene como objetivo principal gestionar la información de los eventos, usuarios y organizadores, proporcionando funcionalidades como:

Gestión de eventos:

- Creación de eventos de manera aleatoria.
- Inscripción y cancelación de inscripciones de usuarios en eventos.
- Confirmación de asistencia a eventos.
- Eliminación y modificación de eventos.

Gestión de usuarios:

- Creación de usuarios aleatorios para simular interacciones con eventos.
- Inscripción a eventos aleatorios.
- Confirmación de asistencia.

Gestión de organizadores:

- Creación de organizadores aleatorios.
- Creación, modificación y eliminación de eventos.

Simulación del portal:

- Generación de eventos con nombres, fechas y ubicaciones aleatorias.
- Interacción simulada mediante un menú en consola.

5.2 TECNOLOGÍAS UTILIZADAS

El backend ha sido desarrollado utilizando:

- **Kotlin:** Lenguaje principal para la lógica de programación.
- **Paradigma de Programación Orientada a Objetos (POO):** Uso de clases para representar eventos, usuarios y organizadores.
- **Estructuras de Datos:** Se han utilizado ArrayList para almacenar eventos y usuarios en la simulación.
- **Generación Aleatoria de Datos:** Se ha implementado un sistema para crear eventos y usuarios de manera automática con datos ficticios.

5.3 CONCLUSIÓN

Dado que el equipo aún no ha implementado una base de datos real, se ha desarrollado una simulación que permite probar la funcionalidad del sistema mediante datos aleatorios. Esto ha permitido validar el funcionamiento del backend y la interacción con los usuarios sin necesidad de conexión a una base de datos.

En futuras versiones se podría realizar la simulación con una integración real con una base de datos SQL, permitiendo un almacenamiento persistente de los eventos y usuarios registrados.

6. DESPLIEGUE DEL PROYECTO EN UN ENTORNO LOCAL

El Portal para la Gestión de Eventos Sostenibles ha sido desplegado en un entorno local utilizando una máquina virtual con Windows 10 y XAMPP como servidor web. Este despliegue permite probar la aplicación sin necesidad de un hosting externo, asegurando su correcto funcionamiento antes de una implementación final.

6.1 CONFIGURACIÓN DE LA MÁQUINA VIRTUAL

Para ejecutar el servidor local, se creó una máquina virtual con Windows 10 Pro utilizando Oracle VirtualBox.

Pasos para la instalación del sistema operativo:

1. Se creó una nueva máquina virtual con las siguientes especificaciones:
 - Nombre: Windows 10 VM.
 - Tipo de sistema: Windows 10 (64-bit).
 - Memoria asignada: 4 GB de RAM.
 - Disco virtual: 50 GB (archivo .vdi en modo dinámico).
2. Se cargó una imagen ISO de Windows 10 en la unidad óptica de VirtualBox para proceder con la instalación.
3. Durante la instalación del sistema operativo:
 - Se seleccionó el idioma: Español (España, internacional).
 - Se eligió la edición de Windows: Windows 10 Pro (x64).
 - Se configuró la distribución de teclado: Español.
 - Se completó la instalación, reiniciando la máquina virtual.

4. Una vez finalizada la instalación de Windows 10, se realizaron ajustes básicos, como:

- Instalación de actualizaciones del sistema.
- Configuración de la cuenta de usuario local.
- Ajustes de red para permitir el acceso al servidor web desde el host.

6.2 INSTALACIÓN Y CONFIGURACIÓN DE XAMPP

Para gestionar el servidor web localmente, se utilizó XAMPP, que incluye Apache, MySQL y PHP.

Pasos de instalación y configuración:

1. Se descargó XAMPP v3.3.0 desde la página oficial de Apache Friends.
2. Se ejecutó el instalador y se configuraron los siguientes módulos:
 - **Apache:** Servidor web.
 - **MySQL:** Base de datos (aunque no se ha utilizado en esta versión).
 - **FileZilla (opcional):** Para futuras pruebas de transferencia de archivos.
3. Se inició XAMPP Control Panel y se activaron los servicios de Apache y MySQL.

Verificación del funcionamiento del servidor:

- Se abrió un navegador y se ingresó la dirección <http://localhost/> para comprobar que el servidor Apache estaba en ejecución.

6.3 IMPLEMENTACIÓN DEL PROYECTO EN XAMPP

Para ejecutar la aplicación web dentro del servidor local, se copiaron los archivos del proyecto en la carpeta htdocs de XAMPP.

Pasos para la implementación:

1. Se ubicó la carpeta htdocs dentro de la instalación de XAMPP (C:\xampp\htdocs).
2. Se copió la carpeta completa del proyecto "**ProyectoIndra**" en htdocs.
3. Se verificó que los archivos HTML y CSS estaban correctamente ubicados.
4. Se accedió a <http://localhost/ProyectoIndra/index.html> en un navegador para probar la carga de la página.

6.4 CONCLUSIÓN

El despliegue del proyecto en un entorno local con XAMPP y una máquina virtual con Windows 10 ha permitido probar el funcionamiento de la aplicación sin necesidad de un servidor en la nube.

Este entorno servirá como base para futuras mejoras, como la conexión con una base de datos y la integración total con el backend en Kotlin.

7. CONCLUSIÓN Y FUTURAS MEJORAS

7.1 CONCLUSIÓN DEL PROYECTO

El desarrollo del Portal para la Gestión de Eventos Sostenibles ha permitido aplicar conceptos fundamentales de desarrollo web y programación en Kotlin, HTML y CSS, además de prácticas esenciales en el despliegue de aplicaciones en un entorno local con XAMPP y VirtualBox.

A lo largo del proceso, se ha logrado:

Implementar una interfaz web funcional y accesible.

Desarrollar un backend en Kotlin que simula la gestión de eventos y usuarios.

Desplegar la aplicación en un entorno virtualizado para su prueba y validación.

Aplicar metodologías de control de versiones y gestión del código en GitHub.

7.2 LIMITACIONES ACTUALES

A pesar de los logros alcanzados, existen algunas limitaciones que se pueden mejorar en futuras versiones:

Falta de una base de datos real: Actualmente: El backend trabaja con datos simulados en memoria. Su implementación permitiría almacenar eventos, usuarios y registros de forma permanente.

Interfaz web más dinámica: Se podrían agregar elementos interactivos con JavaScript para mejorar la experiencia del usuario.

Autenticación de usuarios: Actualmente, no hay un sistema de login real, lo que significa que cualquier usuario puede interactuar con la web sin restricciones.

Conexión entre frontend y backend: El frontend y el backend aún no están integrados de manera real. Actualmente, los eventos se generan aleatoriamente, pero en futuras versiones deberían ser gestionados a través de una API REST.

7.3 FUTURAS MEJORAS

Para una segunda versión del proyecto, se podrían implementar las siguientes mejoras:

Integración con BBDD: Permitiría que los datos se almacenen de manera persistente y que los usuarios puedan inscribirse y gestionar eventos de forma real.

Desarrollo de una API REST: Conectar el frontend con el backend mediante una API REST en Kotlin, facilitando el intercambio de datos.

Uso de JavaScript: Mejorar la interactividad de la interfaz, validaciones de formularios y carga dinámica de eventos.

Autenticación y roles de usuario: Implementar login y permisos para organizadores y usuarios registrados.

Despliegue en un hosting real: Mover la aplicación a un servidor web en la nube para permitir su acceso público.

7.4 REFLEXIÓN FINAL

Este proyecto ha servido como una excelente práctica para consolidar conocimientos en desarrollo web y programación, abordando tanto el frontend como el backend, además de la gestión y despliegue en un entorno controlado.

A medida que se adquieran más conocimientos sobre bases de datos, APIs y autenticación, se podrán implementar nuevas funcionalidades que conviertan este portal en una plataforma completamente funcional y accesible para usuarios reales.