

DOCUMENTO DE PROYECTO

MEMORIA DEL PROYECTO EVENTHUB

Nombre del Proyecto: EventHub

Alumno: Sergio W. Trujillo López

Curso: 2º DAM (Desarrollo de Aplicaciones Multiplataforma)

Asignatura: Proyecto Final de Ciclo

1. Introducción

EventHub es un proyecto web desarrollado como parte del módulo de Proyecto del segundo curso del ciclo formativo de grado superior en Desarrollo de Aplicaciones Multiplataforma (DAM). Se trata de una plataforma de gestión de eventos y conferencias que permite a los usuarios registrarse, comprar tickets electrónicos y recibirlos vía correo electrónico. También incluye un panel de administración para el personal gestor.

Este proyecto combina los conocimientos adquiridos en diferentes módulos: Programación, Acceso a Datos, Desarrollo Web, Servicios y Procesos, y Sistemas de Gestión Empresarial.

2. Tecnologías Utilizadas

- Lenguaje Backend: Python 3 con Flask
- Base de Datos: MySQL (con futura migración a Firebase o Render)
- Frontend: HTML5, CSS3, Bootstrap 5, JavaScript
- Gestor de Dependencias: pip / requirements.txt
- Correo Electrónico: smtplib (Gmail SMTP)
- Generación de PDF: fpdf
- Autenticación y Roles: Sistema propio con sesiones
- Control de Versiones: GitHub (usuario: Strulop)

3. Estructura y Funcionalidades

3.1. Gestión de Usuarios y Roles

- Registro e inicio de sesión con validaciones
- Roles diferenciados: Administrador y Usuario
- Redirecciones según rol

DOCUMENTO DE PROYECTO

3.2. Visualización y Compra de Eventos

- Lista de eventos disponibles (con filtros por categoría y ubicación)
- Vista detallada del evento con información: nombre, fecha, ubicación, duración, descripción, capacidad
- Formulario de compra de tickets con selector de cantidad

3.3. Emisión de Tickets Electrónicos

- Generación automática de ticket en formato PDF con los datos del evento y el comprador
- Estilo profesional con logotipo y colores corporativos
- Envío automático del ticket vía correo electrónico al comprador

3.4. Control de Stock de Tickets

- Verificación de disponibilidad al momento de la compra
- Descuento de tickets adquiridos
- Validación para evitar compras superiores al stock disponible

3.5. Panel de Administración

- CRUD de eventos (crear, editar, eliminar)
- Vista previa de imagen
- Validación de tickets disponibles \leq capacidad
- Filtros por categoría, ubicación y búsqueda textual

4. Arquitectura del Proyecto

- app.py: Archivo principal donde se definen las rutas, controladores y lógica de negocio
- Templates (HTML): home.html, contacto.html, buy_ticket.html, descargar_ticket.html...
- Static (CSS/JS/Img): Archivos de estilo para cada sección, scripts JS para validación y manejo de eventos
- Email_sender.py: Envío de correo con el ticket adjunto (PDF)
- Generador_pdf.py: Creación de ticket en PDF con detalles de la compra
- Base de datos MySQL: Tablas de usuarios, eventos, roles, compras...

DOCUMENTO DE PROYECTO

5. Estilo Visual y Experiencia de Usuario

- Colores institucionales: Azul marino y verde (variables CSS personalizadas)
- Uso de Bootstrap para responsividad y componentes visuales
- Animaciones CSS en la pantalla de descarga del ticket
- Favicon, logos e iconos sociales integrados

6. Consideraciones y Planes Futuros

- Actualmente el sistema utiliza MySQL en entorno local por motivos de simplicidad y facilidad de desarrollo. En una fase de la entrega final, se contempla migrar la base de datos a un servicio gratuito online, como Render, Heroku o Railway, lo cual facilitará el acceso remoto y la escalabilidad del sistema.
- Se podría integrar una pasarela de pago real (Stripe o PayPal)
- Mejoras en el panel de administración con estadísticas y filtros avanzados
- Implementar validación de tickets por QR en la entrada del evento

7. Evaluación, Dificultades y Planteamientos Alternativos

Durante el desarrollo del proyecto [EventHub](#) se han identificado diversos retos técnicos y decisiones que requirieron análisis:

Gestión de roles y sesiones	Fue necesario implementar un sistema robusto que redirigiera correctamente a los usuarios según su rol (administrador o cliente). Esto implicó pruebas con Flask Sessions y condicionales en las rutas.
Validación de tickets:	Asegurar que no se compren más tickets de los disponibles fue uno de los primeros desafíos críticos. Se implementaron restricciones tanto del lado del cliente (JavaScript) como del lado del servidor (Flask).
Generación de PDF y envío de correos	El uso combinado de fpdf para generar tickets visualmente atractivos, y smtpplib para el envío automático, representó una parte importante del trabajo de integración.

DOCUMENTO DE PROYECTO

Diseño adaptable	Se priorizó el diseño responsive mediante Bootstrap para asegurar que el sitio se visualice correctamente en móviles, tablets y escritorios.
Alternativas consideradas	En lugar de MySQL, se contempló desde el inicio Firebase como opción de base de datos para producción. También se evaluó usar Django en lugar de Flask, pero se optó por Flask por su simplicidad y ligereza para un proyecto académico.

Este análisis ha servido para mejorar la toma de decisiones técnicas, y para aprender a afrontar problemas reales como compatibilidad entre librerías, errores de entorno, y diseño centrado en la experiencia del usuario.

8. Conclusión

[EventHub](#) ha sido una oportunidad para poner en práctica gran parte de los conocimientos adquiridos durante el ciclo formativo DAM. He trabajado tanto el frontend como el backend, integrando múltiples tecnologías y librerías. El resultado es una plataforma funcional, escalable y visualmente profesional que cubre un flujo completo desde la publicación de eventos hasta la emisión y entrega del ticket.

Este proyecto me ha permitido mejorar mis habilidades técnicas, comprender mejor la arquitectura de una aplicación web completa y trabajar con servicios externos como correo electrónico o generación de documentos PDF.