

AVANCE II, Agendary

Oscar Eduardo Mercado Guerra,
00113124@uca.edu.sv
Universidad Centroamericana José Simeón Cañas

Resumen — Este trabajo presenta los avances, la implementación y las ideas siguientes de la agenda virtual Agendary, incluyendo la lógica detrás de su estructura, las relaciones de su Backend y Frontend, y las implementaciones que faltan en el programa.

I. INTRODUCCIÓN

El presente documento contiene el Avance II del proyecto Agendary, correspondiente al progreso técnico, investigativo y funcional del desarrollo de la plataforma propuesta. Este informe muestra los avances en backend, frontend, base de datos, pruebas, costos estimados y seguimiento de los objetivos del proyecto.

II. AVANCES DE PROYECTO DE INVESTIGACIÓN

Durante esta fase se evaluó:

A. Avances de la Solución al Problema

El proyecto busca resolver la necesidad de una herramienta sencilla y accesible para gestionar horarios, recordatorios y actividades. Los avances logrados: El diseño de interfaz permite al usuario visualizar sus eventos en un calendario amigable. El backend ya permite registrar, almacenar, consultar y modificar eventos reales. Se implementó un sistema de usuarios independiente, permitiendo sesiones por persona. La base de datos ofrece integridad y seguridad en la gestión de información. Esto demuestra que la solución planteada es viable tanto técnica como operativamente.

B. Refinamiento de la Arquitectura Propuesta

A partir de lo investigado, se definió una arquitectura basada en:
Frontend: HTML, CSS, JavaScript
Backend: Node.js + Express
Base de Datos: SQL Server
Comunicación: API REST JSON

III. PRIMEROS AVANCES DE FRONTEND

A. Pantallas Implementadas

Login
Register
Calendario mensual
Pantalla de creación de eventos

B. Funcionalidades ya operativas

Validación y envío de datos al backend
Renderizado del calendario
Registro local de sesión mediante localStorage
Consumo de API mediante fetch

C. Avances visuales

Diseño limpio y orientado a usabilidad
Estilo responsivo básico
Indicadores visuales de eventos en días específicos

IV. PRIMEROS AVANCES DE BACKEND

TABLA I
AVANCES EN EL BACKEND

Rutas implementadas	
Ruta	Función
/api/register	Registrar usuarios nuevos
/api/login	Validar el usuario ingresado
/api/events	Listar o crear eventos
/api/events/:id	Actualizar o eliminar eventos

A. Funciones Principales

Hash de contraseñas usando **bcrypt**.
Control de errores backend.
Validación de formato de datos entrantes.
Respuestas estructuradas JSON para integrar con frontend.

B. Estado actual del servidor

El servidor Express es completamente funcional, probado y conectado a SQL Server.

* El proyecto será patrocinado completamente por su creador, Oscar Eduardo Mercado Guerra.

V. ESQUEMA DE DATOS

A. Descripción General

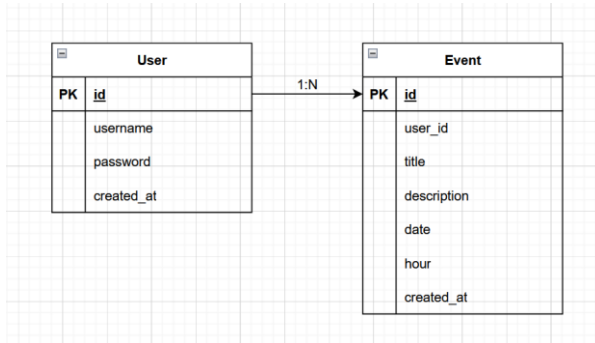
La base de datos está compuesta por dos entidades principales:

Users

↓ 1:N

Events

B. Esquema Entidad-Relación (ER):



VI. SEGUIMIENTO DE COSTOS ESTIMADOS

Los gastos estimados actuales están en la siguiente tabla:

Costos estimados	
Gasto	Costo
Laptop Personal	\$300
Software	\$0
Hosting (Aun no se usará)	\$10/m
Dominio (Aun no se usará)	\$12/a
Total (Anual)	\$432/a

VII. PARTICIPACIÓN TÉCNICA DE LOS INTEGRANTES

A. Oscar Eduardo Mercado Guerra

Responsabilidades Ejecutadas:

Desarrollo completo del backend.

Diseño de base de datos + creación de tablas.

Implementación del frontend (login, register, calendario).

Pruebas de API.

Documentación técnica.

Manejo de repositorio GitHub.

VIII. SEGUIMIENTO DE AVANCES Y OBJETIVOS

A. Objetivos alcanzados

- Backend operativo
- Base de datos funcional
- Login y registro completados
- Integración API-Frontend
- GitHub configurado

B. Objetivos pendientes

- Mejoras en edición visual de eventos
- Optimizaciones de UI/UX
- Fase final de pruebas y QA
- Documentación final del manual de usuario

IX. PLANEACIÓN DE PRUEBAS DE PRODUCTO

A. Pruebas Planeadas

Pruebas funcionales:

Registro, login, navegación del calendario.

Pruebas de integración:

Backend + base de datos, frontend + backend.

Pruebas de rendimiento:

Tiempos de respuesta de API.

Pruebas de validación:

Manejo de errores, campos vacíos, formatos incorrectos.

B. Metodología

Pruebas manuales.

X. ANEXOS

A. Repositorio del Proyecto

Repositorio que contiene los archivos necesarios para hacer uso del programa.

B. Documento de Investigación (Avance I)

Paper inicial que contiene el planteamiento del problema, investigación preliminar, revisión de tecnologías, alcance planteado, diseño de la solución y planificación del proyecto.

XI. REFERENCIAS

- [1] Node.js Documentation, "HTTP & Web Server Modules," *Node.js Foundation*, 2024, <https://nodejs.org/docs/latest/api/>
- [2] Express.js, "Express 5 API Reference," *OpenJS Foundation*, 2024, <https://expressjs.com/>
- [3] Microsoft, "Microsoft SQLServer Documentation," 2024, <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver17>
- [4] bcrypt.js npm Documentation, <https://www.npmjs.com/package/bcryptjs>
- [5] ChatGPT (consultas, limpieza de código), <https://chatgpt.com/>