

# AVANCE I, Agendary

Oscar Eduardo Mercado Guerra,  
[00113124@uca.edu.sv](mailto:00113124@uca.edu.sv)

Universidad Centroamericana José Simeón Cañas

**Resumen —**Este primer trabajo presenta los resultados obtenidos, el avance ejecutado mediante la investigación y la aplicación de actividades orientadas a lograr la implementación de mi proyecto.

## I. PLANTEAMIENTO DEL PROBLEMA

La organización personal de actividades, recordatorios y compromisos es una necesidad cotidiana tanto para estudiantes como para profesionales. Sin embargo, muchas herramientas disponibles presentan interfaces complejas, exceso de opciones o falta de integración básica entre funciones esenciales como gestión de eventos, calendarios y recordatorios simples. Esto provoca que los usuarios pierdan tareas importantes, olviden fechas relevantes o tengan dificultades para visualizar su planificación semanal o mensual. Además, soluciones avanzadas suelen requerir instalación local, licencias de pago o consumo excesivo de recursos.

Por ello, se plantea la creación de Agendary, un sistema web ligero, accesible desde cualquier dispositivo y diseñado específicamente para brindar una plataforma clara y funcional para gestionar eventos personales, con un enfoque en facilidad de uso, acceso inmediato y almacenamiento centralizado.

## II. SOLUCIÓN PLANTEADA

La solución propuesta consiste en desarrollar un sistema web que permita a los usuarios:

- Registrar una cuenta personal.
- Iniciar sesión de forma segura.
- Gestionar un calendario interactivo.
- Crear, editar y eliminar eventos.
- Visualizar sus actividades en formatos diarios, semanales y mensuales.

El sistema opera mediante una arquitectura cliente-servidor, donde:

Frontend: HTML, CSS y JavaScript para ofrecer una interfaz clara y simple.

Backend: Node.js con Express para manejar la lógica del servidor.

Base de datos: SQL Server para almacenar usuarios y eventos.

La aplicación está diseñada para operar sin software adicional más allá de un navegador web moderno y un servidor con Node.js.

## III. DIAGRAMAS TÉCNICOS INICIALES DE LA SOLUCIÓN

### A. Arquitectura General

Cliente: Interfaz web que permite interacción directa con funciones del calendario.

Servidor: API REST que maneja solicitudes como login, registro y gestión de eventos.

Base de datos: Tablas Users y Events estructuradas para soportar consultas eficientes.

### B. Flujo General del Usuario

El usuario ingresa al sitio y se registra.

El sistema almacena la información en la base de datos.

Luego el usuario inicia sesión.

El backend valida credenciales.

El frontend crea una sesión local y redirige al calendario.

El usuario gestiona sus eventos.

La información se almacena y se muestra en el calendario.

### C. Diagrama Entidad–Relación

Entidades:

Users: id, username, password\_hash, created\_at

Events: id, user\_id, title, description, date, hour, created\_at

Relación: Un usuario puede tener múltiples eventos (1:N).

## IV. PLANIFICACIÓN DE LA EJECUCIÓN

La planificación se dividió en cuatro etapas:

### Etapa I - Investigación y Análisis

Identificación del problema

Definición de objetivos

Selección de tecnologías

Diseño inicial del sistema

Definición del modelo de datos

### Etapa II - Desarrollo Inicial

Creación del backend y API

Configuración de la base de datos

Primer prototipo de pantallas

Implementación inicial del login y registro

### Etapa III - Desarrollo del Producto Final

Integración completa del calendario

Ajustes de interfaz

Pruebas y correcciones

### Etapa IV - Cierre y Documentación

Elaboración de manuales

Paper final

Publicación del proyecto

---

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

## V. EQUIPO INVOLUCRADO

El desarrollo del proyecto es realizado por un único estudiante:

Oscar Eduardo Mercado Guerra, carnet 00113124  
 Funciones principales: análisis, diseño, creación del frontend, creación del backend, implementación del frontend, implementación del backend, creación, diseño y gestión de la base de datos, documentación y despliegue.

## VI. COSTOS ESPERADOS DEL PROYECTO

Como proyecto académico, se determinan los siguientes costos aproximados:

### A. Recursos Técnicos

Laptop personal  
 Conexión a internet

### B. Software libre

Node.js  
 Visual Studio Code  
 SQL Server

### C. Tiempo Invertido

Investigación: 3 horas  
 Diseño: 2 horas  
 Desarrollo: 12+ horas

### D. Costos Opcionales (si fuera implementación real)

Hosting web: \$5–10 mensuales  
 Dominio: \$12 anuales

Costo: \$0 (uso de herramientas gratuitas y personales).

## VII. TECNOLOGÍAS A UTILIZAR

### A. Frontend

HTML  
 CSS  
 JavaScript  
 Diseño responsivo

### B. Backend

Node.js  
 Express.js  
 Bcrypt (protección de contraseñas)

### C. Base de Datos

SQL Server  
 Conexión mediante mssql (Node)

### D. Control de Versiones

Git y GitHub

## VIII. BIBLIOGRAFÍA

Mozilla Developer Network (MDN Web Docs). *JavaScript, HTML y APIs del navegador.*

Microsoft Learn. *Documentación de SQL Server y autenticación en bases de datos.*

Express.js Official Documentation.

GitHub Documentation. *Version control and repositories.*

ChatGPT. *Asistencia técnica en limpieza del código y en creación del sistema.*