

Agendary: Sistema Web de Gestión Personal de Eventos y Calendario Interactivo

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

Resumen— El presente documento presenta el diseño, análisis, desarrollo e implementación de *Agendary*, una aplicación web de organización personal basada en un calendario interactivo y un sistema de usuarios con autenticación. El objetivo del proyecto es ofrecer una herramienta sencilla y accesible que permita gestionar actividades diarias, visualizar eventos por categoría y exportar información para consulta externa. La solución se desarrolló bajo un enfoque modular, empleando tecnologías web estándar y una arquitectura basada en un backend en Node.js conectado a una base de datos SQL Server.

Índice de términos - Frontend: Parte visual del sistema, donde interactúa el usuario, Backend: Lógica del servidor que procesa peticiones y gestiona datos, CORS: Permisos de intercambio de recursos entre dominios, API: Conjunto de rutas expuestas por el servidor para comunicar frontend y backend, SQL Server: Motor de base de datos utilizado en el proyecto, bcrypt: Librería para cifrar contraseñas, GitHub: Plataforma de control de versiones y almacenamiento del proyecto.

I. INTRODUCCIÓN

La gestión del tiempo es una necesidad fundamental en el ámbito académico, laboral y personal. Con frecuencia, los usuarios recurren a aplicaciones dispersas que no integran herramientas de planificación visual, categorización y exportación de actividades. *Agendary* surge como una solución unificada que centraliza la administración de eventos mediante un calendario mensual dinámico.

II. OBJETIVOS DEL PROYECTO

A. Objetivo General

Desarrollar un sistema web funcional que permita a los usuarios gestionar recordatorios, eventos y actividades personales mediante una interfaz intuitiva y un backend seguro.

B. Objetivos Específicos

Implementar un sistema de autenticación que permita registrar y gestionar usuarios con contraseñas cifradas.
Diseñar una interfaz sencilla y accesible para el manejo de recordatorios.
Asegurar que el sistema pueda funcionar en múltiples dispositivos.

III. ANÁLISIS DE REQUERIMIENTOS

A. Determinación del Alcance del Proyecto

El sistema permitirá:

Registrar usuarios.

Iniciar sesión.

Crear, editar y eliminar recordatorios.

Visualizarlos en una interfaz central.

Almacenar información en una base de datos segura.

B. Obtención y Negociación de Requisitos

Los requisitos derivan de necesidades básicas de organización:

Los usuarios deben poder gestionar sus actividades de forma sencilla.

La autenticación debe ser segura.

C. Perspectiva del Producto (UI)

La interfaz mantiene una estética centrada, minimalista, con formularios limpios y botones visibles. Incluye:

Pantalla de login

Pantalla de registro

Agenda principal con formulario y lista de actividades

D. Características de Usuarios

Estudiantes

Profesionales

Usuarios que requieren orden personal

No requiere conocimientos técnicos avanzados.

IV. FACTIBILIDAD DEL PROYECTO

A. Factibilidad Técnica

Tecnologías disponibles:

Node.js

Express

SQL Server

HTML, CSS y JavaScript

B. Factibilidad Operativa

El sistema es simple y fácil de usar. No requiere instalación por parte del usuario final.

C. Factibilidad Económica

El desarrollo utiliza software gratuito. El costo principal es el tiempo del desarrollador.

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

V. DISEÑO DEL SISTEMA

A. Arquitectura General

El sistema utiliza una arquitectura basada en cliente-servidor con Flask como backend y SQL Server para el almacenamiento de datos.

B. Componentes Principales

Frontend: Interfaz desarrollada en HTML, CSS y JS.

Backend: API REST en Flask.

Base de Datos: Modelo relacional optimizado para consultas rápidas.

C. Flujo de Datos

El usuario interactúa con el frontend, que envía peticiones al backend. Este procesa la solicitud y consulta o modifica los datos en SQL Server.

VI. DESARROLLO E IMPLEMENTACIÓN

A. Implementación

Frontend ligero con HTML/CSS/JS.

Backend Express con rutas /api/login y /api/register.

SQL Server conectado mediante mssql.

B. Control de Versiones

Repositorio GitHub con commits organizados y .gitignore para no subir/descargar todos los archivos con cada push.

C. Confiabilidad, disponibilidad y mantenimiento

El sistema depende del servidor, pero es estable localmente. Arquitectura modular facilita mantenimiento.

D. Documentación de Pruebas

Incluye pruebas:

Login correcto/incorrecto

Registro duplicado

Creación de recordatorios

E. Plan de Capacitación

Uso del login

Gestión de actividades

Explicación rápida de flujo general

F. Plan de Mantenimiento

Limpieza periódica de la base

Verificación de integridad de datos

Actualización de dependencias

VII. MEJORAS FUTURAS

A. Implementación de Notificaciones "Push"

Se planea integrar notificaciones en tiempo real para dispositivos móviles.

B. Módulo de Colaboración

Permitirá compartir calendarios y eventos entre usuarios.

C. Integración con Servicios Externos

Como Google Calendar o Microsoft Outlook.

VIII. ANEXOS

A. Manual Técnico del Sistema

Documento en formato PDF que describe la arquitectura del sistema, configuración del servidor, estructura de la base de datos, API, dependencias y el proceso de despliegue.

B. Manual de Usuario

Guía ilustrada con instrucciones para el uso del sistema

Agendary, incluyendo registro, inicio de sesión, creación de eventos y navegación de la interfaz.

C. 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.

D. Reporte de Avance de Proyecto (Avance II)

Informe que documenta el progreso del desarrollo, primeras implementaciones funcionales, estructura del backend, prototipo del frontend, esquema ER y evidencia del repositorio.

E. Repositorio del Proyecto

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

IX. REFERENCIAS

Se muestran a continuación los diferentes enlaces relacionados con la elaboración de este Proyecto.

- [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/>
- [6] Mozilla Developer Network, "JavaScript Guide and Web APIs," *MDN Web Docs*, 2024, <https://developer.mozilla.org>
- [7] IEEE, "IEEE Author Center: Manuscript Templates for Conference Proceedings," *Institute of Electrical and Electronics Engineers*, 2024. Disponible en: <https://www.scribbr.com/ieee/ieee-website-citation>