

Informe de Diseño de Software

Proyecto: Sistema de Gestión de Ciudadanos Intergalácticos - Interpolice

Formato con Normas APA

Introducción

El presente informe presenta el diseño del sistema de información para Interpolice, encargado de registrar y consultar información de ciudadanos intergalácticos y su historial de delitos. El diseño se enfoca en una arquitectura orientada a servicios (SOA), utilizando API REST, con enfoque en la modularidad, escalabilidad, y mantenibilidad.

Objetivos del diseño

- Definir la arquitectura del sistema.
 - Describir los componentes funcionales.
 - Modelar el comportamiento del sistema mediante diagramas.
 - Establecer la estructura de datos relacional normalizada.
 - Diseñar una interfaz amigable con mockups propuestos.
-

1. Arquitectura propuesta

- **Tipo:** Orientada a Servicios (SOA)
- **API REST:** Se implementará una API en Node.js que permita el acceso a recursos a través de endpoints.
- **Cliente web:** Interfaz creada con HTML, CSS (Framework) y Vanilla JavaScript estrictamente.
- **Controlador por módulo:** Se aplicará el patrón MVC en los módulos de la API (sin orientación a objetos).
- **Seguridad:**
 - Autenticación con roles: General, Policía, Secretaria, Administrador.
 - Autenticación de 2 factores.
 - Protección de endpoints con tokens JWT.

2. Diagramas

2.1. Diagrama de casos de uso extendidos

(Incluir imagen del diagrama de casos de uso donde se visualicen: login, registrar ciudadano, consultar antecedentes, registrar delitos, emitir amonestaciones, etc.)

2.2. Diagrama de actividades

(Mostrar flujo de: registrar ciudadano, registrar delito, emitir amonestación)

2.3. Diagrama de componentes

(Representar la interacción entre los componentes: cliente web, API, base de datos, servicios de autenticación)

2.4. Modelo Entidad-Relación (ER)

(Base de datos relacional normalizada con entidades: Ciudadano, Delito, Usuario, Amonestacion, etc.)

3. Mockup de interfaz

Como anexo se incluye el mockup de la interfaz donde los usuarios (policías, generales) pueden consultar, registrar y gestionar la información.

4. Tecnologías a utilizar

- **Frontend:** Vanilla JS, CSS (Tailwind o Bootstrap), SweetAlert para retroalimentación.
 - **Backend:** Node.js, Express, arquitectura REST.
 - **Base de Datos:** MySQL.
 - **Pruebas:** Jest (unitarias), Postman (API), Selenium (interfaz).
 - **Control de versiones:** Git y Github.
-

5. Justificación de diseño

El sistema debe funcionar en tablets y dispositivos antiguos (iPhone 47 XPlus Sigma), por lo tanto se diseñó una interfaz ligera, responsiva y accesible. La arquitectura modular y orientada a servicios facilita escalar funciones como exportación a Excel, generación de gráficas y protección por roles.

Referencias (APA)

IEEE. (1998). IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998).

(Agregar cualquier otra referencia a material del SENA, manuales técnicos, o fuentes académicas utilizadas).