



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Universidad Nacional de Colombia - sede Bogotá

Facultad de Ingeniería

Departamento de Sistemas e Industrial

Curso: Ingeniería de Software 1 (2016701)

Estudiantes: Juan Camilo D'Aleman Rodriguez, David

Ladino, Jorge Cuadrado, Smith Yañez.

Colombia Raíces



Colombia Raíces es una aplicación de escritorio diseñada para promover el turismo comunitario en regiones marginadas de Colombia, conectando viajeros con experiencias auténticas ofrecidas por comunidades indígenas, campesinas y locales en destinos poco conocidos, como La Guajira, Chocó, Amazonía y pueblos patrimonio (e.g., Barichara, Mompox). La aplicación permite a los usuarios explorar experiencias turísticas (talleres culturales, caminatas guiadas, visitas históricas), generar estimaciones de reservas y conocer comunidades locales, mientras que los operadores comunitarios pueden publicar sus experiencias. La plataforma busca visibilizar regiones marginadas, fomentar el turismo sostenible y apoyar el desarrollo económico de estas comunidades.

Problema: Muchas regiones de Colombia, ricas en patrimonio cultural, histórico y natural, carecen de visibilidad turística debido a su marginalidad geográfica o social. Esto limita las oportunidades económicas para las comunidades locales y perpetúa la desigualdad. Las plataformas turísticas existentes suelen centrarse en destinos comerciales, ignorando el potencial de experiencias auténticas en comunidades indígenas o rurales.

Justificación: Colombia Raíces aborda un problema realista y acotado al promover el turismo comunitario en regiones marginadas, ofreciendo una solución tecnológica que conecta viajeros con experiencias auténticas. El proyecto cumple con las restricciones de la asignatura al implementarse como una arquitectura monolítica, usar una base de datos relacional (SQLite) y seguir el ciclo completo de desarrollo de software, incluyendo análisis de requerimientos, diseño, desarrollo, pruebas y documentación.

1. Objetivos

Objetivo General

Desarrollar una aplicación de escritorio monolítica que promueva el turismo comunitario en regiones marginadas de Colombia, permitiendo a los usuarios explorar experiencias turísticas, generar estimaciones de reservas y conectar con comunidades locales, siguiendo principios de ingeniería de software (Clean Code, patrones de diseño, testing).

Objetivos Específicos

1. Diseñar e implementar una aplicación con arquitectura monolítica usando Electron, JavaScript, Tailwind CSS y SQLite.
2. Permitir a los usuarios buscar y filtrar experiencias turísticas por región, tipo (cultural, histórica, ecológica) y presupuesto.
3. Habilitar la generación de estimaciones de reservas basadas en reglas de negocio definidas.
4. Permitir a operadores comunitarios (o administradores) publicar experiencias con descripciones, fotos y precios estimados.
5. Integrar mapas interactivos (usando OpenStreetMap) como funcionalidad complementaria para mostrar ubicaciones.
6. Aplicar el ciclo completo de desarrollo de software, incluyendo análisis de requerimientos, diseño con patrones, desarrollo, pruebas y documentación.

2. Alcance

Funcionalidades Principales:

1. Registro y autenticación de usuarios (viajeros y operadores comunitarios).
2. Búsqueda y filtrado de experiencias turísticas por región, tipo y presupuesto.
3. Generación de estimaciones de reservas con cálculos basados en duración, tipo de experiencia y servicios.
4. Publicación y validación de experiencias por parte de operadores comunitarios (con revisión administrativa simulada).
5. Visualización de perfiles de comunidades (historia, cultura, fotos).
6. Mapas interactivos para mostrar ubicaciones de experiencias (usando OpenStreetMap).

Limitaciones:

1. No se procesan pagos ni se realizan reservas reales.
2. El proyecto se centra en 3-5 regiones marginadas (e.g., La Guajira, Chocó, Amazonía, Barichara).
3. No se exponen APIs públicas; las APIs externas (e.g., OpenStreetMap) son complementarias.

4. La aplicación es de escritorio, no web ni móvil.

3. Requerimientos

Requerimientos Funcionales

RF1: El sistema debe permitir a los usuarios registrarse y autenticarse con email y contraseña.

RF2: El sistema debe permitir a los viajeros buscar experiencias por región, tipo (cultural, histórica, ecológica) y presupuesto.

RF3: El sistema debe generar estimaciones de reservas basadas en la duración, tipo de experiencia y servicios incluidos.

RF4: El sistema debe permitir a los operadores comunitarios subir experiencias (nombre, descripción, fotos, precio estimado) con un proceso de validación administrativa.

RF5: El sistema debe mostrar perfiles de comunidades con información cultural e histórica.

RF6: El sistema debe integrar mapas interactivos para mostrar ubicaciones de experiencias.

Requerimientos No Funcionales

RNF1: La aplicación debe tener una arquitectura monolítica implementada con Electron, JavaScript, Tailwind CSS y SQLite.

RNF2: La interfaz debe ser visualmente atractiva, usando Tailwind CSS con un diseño inspirado en la diversidad cultural de Colombia.

RNF3: El código debe seguir principios de Clean Code (nombres descriptivos, funciones cortas, modularidad).

RNF4: El sistema debe incluir pruebas unitarias.

RNF5: La aplicación debe ser portable y fácil de instalar en sistemas Windows, macOS y Linux.

RNF6: La base de datos debe garantizar integridad referencial y estar normalizada.

4. Reglas De Negocio

Las siguientes reglas de negocio establecen las restricciones y comportamientos clave para garantizar el cumplimiento de los requerimientos funcionales (RF1-RF6) y la coherencia del sistema.

Registro y Autenticación de Usuarios:

Los usuarios deben registrarse con un nombre, email único, contraseña y tipo de usuario (viajero u operador comunitario).

La contraseña debe tener al menos 8 caracteres, incluir al menos una letra (mayúscula o minúscula) y un número.

El email debe ser único en la base de datos (tabla Usuarios).

Solo los usuarios registrados pueden autenticarse, y la autenticación requiere un email existente y una contraseña correcta.

El sistema no debe revelar si el error de autenticación es por email o contraseña, usando un mensaje genérico (“Email o contraseña incorrectos”).

La sesión permanece activa hasta que el usuario cierre la aplicación o seleccione la opción de cerrar sesión.

Búsqueda de Experiencias Turísticas:

Solo las experiencias con estado “aprobada” en la tabla Experiencias son visibles en las búsquedas.

Los filtros de búsqueda (región, tipo de experiencia, presupuesto máximo) son opcionales; si no se especifican, se muestran todas las experiencias aprobadas.

El tipo de experiencia debe ser uno de: cultural, histórica o ecológica.

Gestión de Reservas:

Las reservas solo pueden generarse para experiencias aprobadas.

La fecha de una reserva debe ser futura (posterior a la fecha actual).

El costo estimado de una reserva se calcula sumando el precio base de la experiencia y porcentajes adicionales por servicios (+20% por guía, +15% por transporte, por ejemplo).

Solo las reservas en estado “pendiente” o “confirmada” pueden ser canceladas.

Las reservas están asociadas a un usuario específico y no pueden ser modificadas por otros usuarios.

Publicación y Aprobación de Experiencias:

Las experiencias deben incluir nombre, descripción, tipo (cultural, histórica, ecológica), precio estimado mayor a 0, duración y fotos.

Las fotos de experiencias deben ser en formato JPEG o PNG y menores a 5 MB.

Las experiencias nuevas se registran con estado “pendiente” y requieren aprobación administrativa para cambiar a “aprobada”.

Los cambios en campos críticos de una experiencia (e.g., precio, descripción) cambian su estado a “pendiente” para revalidación.

Solo los operadores comunitarios propietarios de una experiencia pueden actualizarla.

Gestión de Comunidades:

Los perfiles de comunidades deben incluir nombre, región, descripción (mínimo 50 caracteres) y al menos una foto en formato JPEG o PNG, menor a 5 MB.

Las comunidades deben estar registradas en la tabla Comunidades antes de asociarse con experiencias.

Mapas Interactivos:

Las ubicaciones de las experiencias deben estar asociadas a experiencias aprobadas y almacenadas como coordenadas (latitud, longitud) en la tabla Experiencias.

Los mapas interactivos solo muestran experiencias aprobadas.

Actualización de Perfil de Usuario:

El email de un usuario no puede modificarse tras el registro.

Las actualizaciones de contraseña deben cumplir con las mismas reglas de validación que el registro (mínimo 8 caracteres, al menos una letra y un número).

Seguridad y Privacidad:

Las contraseñas se almacenan encriptadas (usando bcrypt) y nunca en texto plano.

Solo los usuarios autenticados pueden acceder a funcionalidades protegidas (e.g., reservas, publicación de experiencias).

Solo el administrador puede registrar comunidades y aprobar o rechazar experiencias.

Los datos sensibles (e.g., sesiones, contraseñas) deben eliminarse de la memoria al cerrar sesión.

Se implementa un límite de intentos de inicio de sesión para prevenir ataques de fuerza bruta.

Integridad de Datos:

Las experiencias deben estar asociadas a una comunidad existente en la tabla Comunidades.

Las reservas deben estar asociadas a un usuario y una experiencia existentes en las tablas Usuarios y Experiencias, respectivamente.

Las validaciones (e.g., formato de fotos, fechas futuras, precios) se realizan en el backend para garantizar consistencia.