



1. Lenguaje de programación y framework seleccionado

 Lenguaje elegido: Java

 Framework de interfaz gráfica: JavaFX

 Comparativo de tecnologías

Tecnología	Ventajas	Desventajas
Java + JavaFX	<ul style="list-style-type: none">- Amplio soporte multiplataforma- Fácil integración con JDBC y PostgreSQL- Separación clara con patrón MVC- Interfaz rica con soporte de multimedia	<ul style="list-style-type: none">- Interfaz menos moderna en apariencia comparado con frameworks web- Curva de aprendizaje inicial
Python + Tkinter	<ul style="list-style-type: none">- Más fácil de aprender- Menos código- Buen prototipado rápido	<ul style="list-style-type: none">- Pobre escalabilidad- Interfaces más limitadas y menos profesionales
Electron (JS + HTML)	<ul style="list-style-type: none">- Interfaz moderna- Reutiliza conocimientos web	<ul style="list-style-type: none">- Alto consumo de recursos- Mayor complejidad de configuración
C# + WPF	<ul style="list-style-type: none">- UI potente en Windows- Integración con Visual Studio	<ul style="list-style-type: none">- Dependiente de Windows- Menor portabilidad

Justificación:

Se eligió **Java con JavaFX** porque:

- El curso está centrado en **buenas prácticas de programación orientada a objetos**, y Java es excelente en este enfoque.

- Permite una estructura clara usando el patrón **Modelo-Vista-Controlador (MVC)**, facilitando la colaboración por capas.
- JavaFX soporta **interfaces interactivas y personalizables** (botones, listas, temas, imágenes, gráficos).
- El equipo tiene experiencia previa con Java y ha trabajado en otros proyectos con esta tecnología.
- Es una solución **multiplataforma** compatible con Windows, Linux y macOS, ideal para un diario de viajes.

2. Base de datos relacional utilizada

Tecnología seleccionada: PostgreSQL

✓ Comparativo con otras bases de datos

Base de datos	Ventajas	Desventajas
PostgreSQL	<ul style="list-style-type: none"> - Muy robusta y profesional - Soporte para integridad referencial - Alto rendimiento - Multiplataforma 	<ul style="list-style-type: none"> - Más compleja de configurar que SQLite
SQLite	<ul style="list-style-type: none"> - Súper ligera - No requiere servidor - Ideal para apps pequeñas 	<ul style="list-style-type: none"> - No multiusuario - Menos robusta
MySQL	<ul style="list-style-type: none"> - Amplia documentación - Similar a PostgreSQL 	<ul style="list-style-type: none"> - Menos orientada a estándares SQL
MongoDB (NoSQL)	<ul style="list-style-type: none"> - Alta escalabilidad y flexibilidad 	<ul style="list-style-type: none"> - No adecuada para relaciones complejas

Justificación:

Se seleccionó PostgreSQL por ser:

Una base de datos relacional completa, confiable y gratuita.

Compatible con JDBC para integración directa con Java.

Soporta integridad referencial, relaciones complejas y tipos de datos robustos.

Ideal para una aplicación como Bogotravel, que requiere múltiples tablas (usuarios, entradas, lugares, fotos, etc.) con relaciones claras.

3. Bibliotecas y herramientas complementarias

Herramienta / Librería	Función principal
JavaFX	Crear interfaces graficas de usuario
PostgreSQL JDBC Driver	Conexión entre Java y la base de datos PostgreSQL
BCrypt (jbcrypt)	Hash seguro de contraseñas
Maven	Gestión de dependencias y construcción del proyecto
DB Browser for PostgreSQL	Visualización y administración de la base de datos
IntelliJ IDEA	Entorno de desarrollo (IDE) principal
Java 17+	Lenguaje principal, versión moderna y estable

4. Justificación detallada de las elecciones

Relación con los objetivos del curso

El curso busca fomentar buenas prácticas de desarrollo (modularidad, separación por capas, documentación, pruebas).

Java y PostgreSQL permiten aplicar conceptos clave como:

Patrones de diseño (DAO, MVC)

Acceso a bases de datos relacionales

Organización de código en paquetes

Uso de herramientas reales de desarrollo profesional (Maven, IDEs)

Capacidades del equipo

El equipo tiene experiencia previa con Java y bases de datos SQL.

Se busca que el equipo pueda trabajar en paralelo:

Un integrante en el backend

Otro en la interfaz con JavaFX

Otro en documentación y pruebas

Las herramientas elegidas son gratuitas, bien documentadas y estables, lo que facilita el trabajo colaborativo.

Necesidades del proyecto

Se requiere una app que funcione sin conexión a Internet, por eso no se usó una base web ni frameworks como React o Angular.

Se necesitaba una interfaz amigable y personalizada, lo que permite JavaFX con temas visuales, imágenes y diseño responsivo.

Se requería seguridad básica de usuarios, por eso se integró BCrypt para cifrar contraseñas.