

Memoria TFG - IslandesApp

IslandesApp

1. Introducción

IslandesApp es una aplicación de escritorio desarrollada en Java cuyo objetivo es facilitar el aprendizaje del idioma islandés mediante lecciones organizadas, vocabulario clasificado y ejemplos prácticos.

Este proyecto ha sido desarrollado como Trabajo de Fin de Grado (TFG) por **James Francis Bastian** dentro del programa del Máster en Desarrollo Full Stack del **Grupo Atrium**.

2. Objetivos

El objetivo principal es crear una herramienta práctica que permita gestionar y consultar lecciones y vocabulario en islandés para apoyar el aprendizaje autodidacta del idioma.

Además, se busca aplicar conocimientos técnicos adquiridos durante el máster, incluyendo el diseño modular, persistencia con MongoDB, patrones de diseño y pruebas unitarias.

3. Arquitectura y Diseño

La aplicación está diseñada siguiendo el patrón MVC (Modelo-Vista-Controlador) para separar la lógica de negocio, la interfaz y el acceso a datos.

Para la persistencia se utiliza **MongoDB**, accediendo a los datos a través de **DAOs específicos** para las entidades principales: *Lección* y *Palabra*.

El proyecto está estructurado en capas:

- **Modelo:** Clases Leccion y Palabra.
- **DAO:** Clases que acceden a MongoDB.
- **Servicio:** Lógica de negocio.
- **Vista:** Interfaz gráfica en Swing.
- **Tests:** Pruebas unitarias con JUnit.

4. Tecnologías utilizadas

- Java 17
- MongoDB
- Maven como gestor de dependencias y compilación
- JUnit 5 para pruebas unitarias
- Swing para la interfaz gráfica
- Eclipse IDE para desarrollo

5. Instrucciones de uso

1. Asegúrate de tener MongoDB instalado y en ejecución en localhost:27017.

2. Clona el repositorio del proyecto: `git clone https://github.com/tu_usuario/IslandesApp.git`
3. Abre el proyecto con tu IDE favorito (por ejemplo, Eclipse).
4. Ejecuta la clase `MainApp.java` para lanzar la aplicación.
5. Usa la interfaz para explorar las lecciones y palabras, añadir contenido y gestionar el vocabulario.

6. Resultados y Conclusiones

IslandesApp ha cumplido con los objetivos planteados, proporcionando una herramienta práctica para el aprendizaje del islandés mediante la gestión estructurada de contenidos. La arquitectura modular y el uso de tecnologías actuales garantizan que la aplicación sea escalable, mantenible y susceptible de futuras ampliaciones, como la inclusión de ejercicios interactivos o integración con servicios externos.

La experiencia personal de crear esta aplicación mientras aprendía islandés ha sido muy enriquecedora y ha permitido aplicar conocimientos técnicos en un proyecto real y útil.

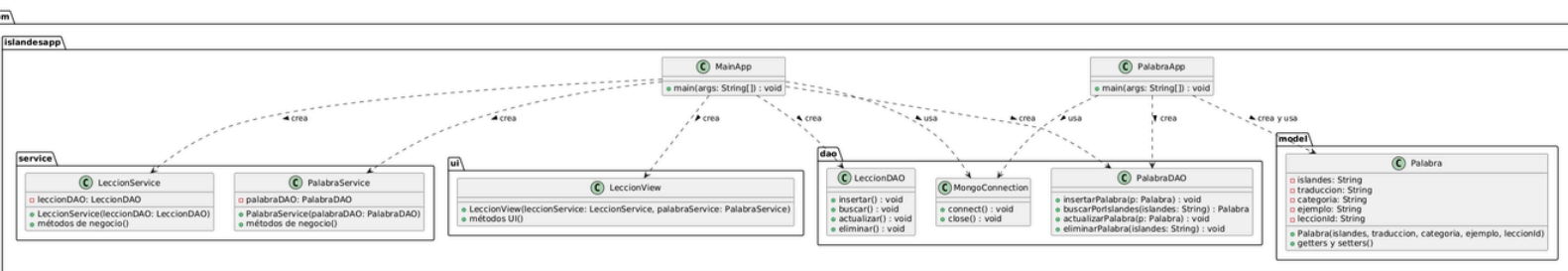
7. Bibliografía

- Documentación oficial de MongoDB: <https://docs.mongodb.com/>
- Documentación oficial de Java: <https://docs.oracle.com/en/java/>
- Tutoriales y guías sobre Java Swing
- Materiales sobre patrones de diseño MVC y DAO
- Recursos de aprendizaje de idiomas

8. Anexos

- Código fuente completo disponible en el repositorio
- Diagrama UML (ver más abajo)
- Resultados de las pruebas unitarias
- Archivo `README.md` con instrucciones de uso y estructura del proyecto

Diagrama UML del sistema (IslandesApp)



Plan de Pruebas - IslandesApp

1. Introducción

Objetivo del plan de pruebas:

Garantizar que todas las funcionalidades de IslandesApp funcionan correctamente según los requisitos.

Alcance del proyecto:

- Gestión de lecciones y palabras.
- Interacción con la base de datos MongoDB.
- Funcionalidades básicas de búsqueda y validación de datos.

2. Entorno de pruebas

- Herramientas usadas: JUnit para pruebas unitarias.
- Base de datos MongoDB en entorno local o de pruebas.
- IDE recomendado: IntelliJ IDEA o Eclipse.
- Versión de Java: 17 o superior.
- Configuración de conexión a MongoDB necesaria.

3. Tipos de pruebas

- **Pruebas unitarias:** Validar métodos de servicios y DAO.
- **Pruebas de integración:** Verificar la conexión y operaciones con MongoDB.
- **Pruebas funcionales:** Comprobar la correcta ejecución de casos de uso.

4. Casos de prueba

- Crear lección: Validar que se crea correctamente una nueva lección.
- Modificar lección: Comprobar que se actualizan los datos correctamente.
- Eliminar lección: Verificar la eliminación efectiva.
- Añadir palabra a lección: Probar la asociación correcta.
- Buscar palabra: Confirmar resultados correctos para búsquedas.
- Validar conexión con MongoDB: Asegurar conexión exitosa.
- Comprobar validación de datos: Validar manejo de entradas inválidas.

5. Criterios de aceptación

- Las operaciones CRUD deben ejecutarse sin errores.
- Los datos deben persistir correctamente en MongoDB.
- La interfaz debe mostrar mensajes claros de éxito o error.
- El sistema debe manejar entradas inválidas con mensajes de error adecuados.

6. Responsables y fechas

- **Responsable del plan de pruebas:** James Bastian

- **Fecha estimada de finalización de pruebas:** 23 de mayo de 2025
- **Revisión y actualización continua durante el desarrollo.**