PROYECTO INTEGRADOR AVANCE #1
25 Agosto 2025
Integrantes: Jatnael Tigrero, Josue Vivas

Functional requirements

1. User management

RF1: The system must allow teachers and administrators to register and log in.
RF2: The system must manage roles (administrator, teacher, academic coordinator).

2.Curriculum development

RF3: The system must allow for the creation, editing, and deletion of religious education curriculum content.

RF4: The system must allow for the collaborative co-creation of the area plan (several users working on the same plan).

RF5: The system must allow for the organization of information into academic levels (primary, secondary, middle school).

RF6: The system must automatically generate the curriculum based on the content entered.

3. Document generation

RF7: The system must generate a digital document (PDF/Word) with the area plan and curriculum.

RF8: The system must allow the curriculum document to be downloaded and printed.

4. Access and consultation

RF9: The system must allow users to search and consult curriculum content by level, grade, or subject.

RF10: The system must offer a simple and intuitive interface to facilitate lesson planning.

Non-functional requirements

1.Usability

RNF3: The platform must be intuitive and easy to use for teachers without advanced technological experience.

RNF4: It must have a user manual and basic tutorials.

2.Security

RNF5: The system must ensure that only authorized users can modify the area plan.

3.Maintainability

RNF8: The code must be documented and follow clean development standards.

User Stories

User Story #1

Number: 1
User: Teacher
Story Name: Create religion area plan
Business Priority: High
Development Risk: Medium
Estimated Points: 4 (NORMAL complexity → FTRs = 2, DETs = 6)
Assigned Iteration: 1
Responsible Developer: To be assigned

Description:
As a teacher, I want to create a religion area plan by selecting grades, contents, and competencies to build the curriculum map.

Validation:
The teacher can save a new area plan, and it remains available in the system.

User Story #2

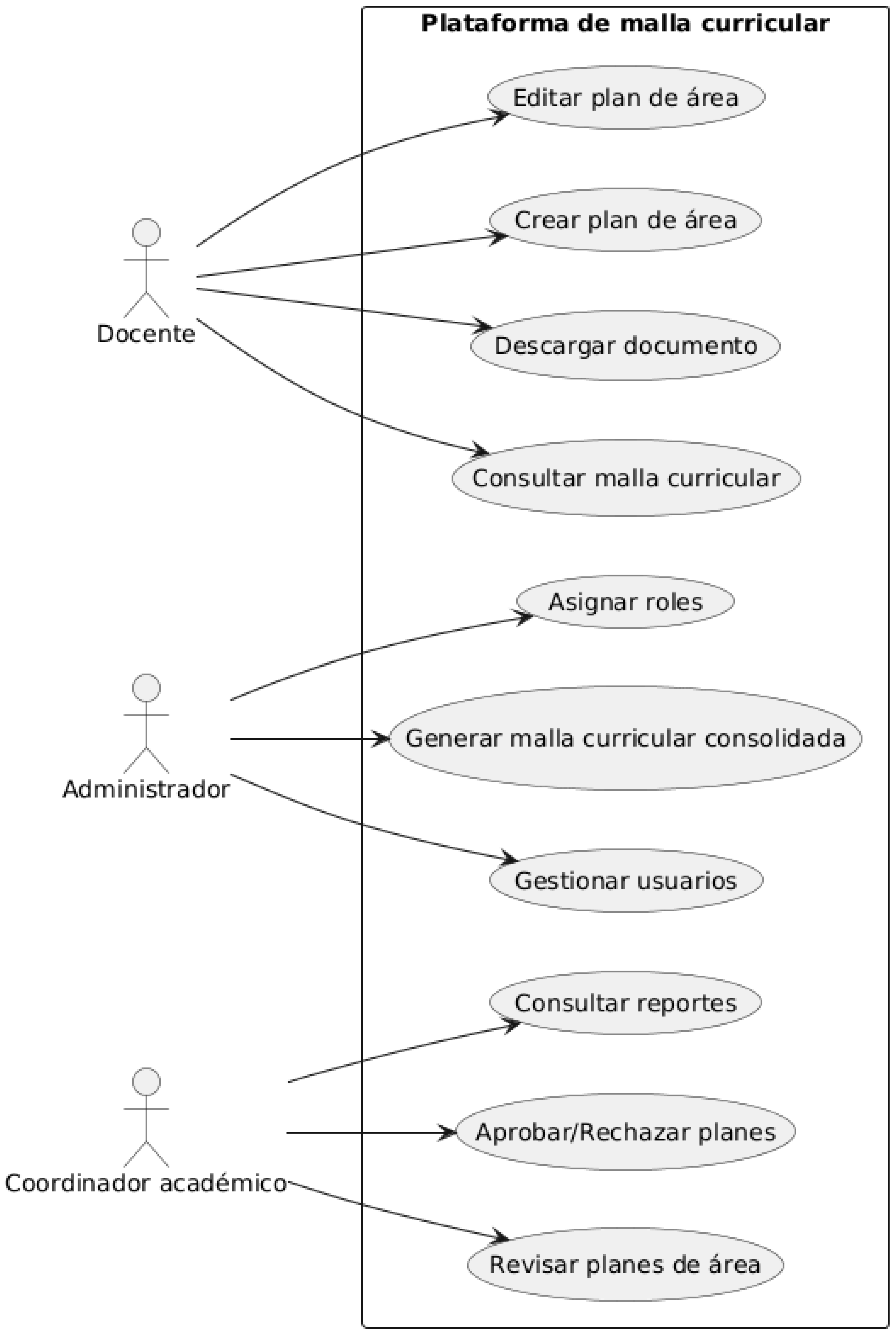
Number: 2
User: Teacher
Story Name: Edit area plan contents
Business Priority: Medium
Development Risk: Low
Estimated Points: 3 (LOW complexity → FTRs = 1, DETs = 4)

Assigned Iteration: 2
Responsible Developer: To be assigned

Description:
As a teacher, I want to edit the contents of an existing area plan so I can update competencies, objectives, or activities.

Validation:
The system allows modifying an existing area plan and saves the changes into the database.

Use case diagram

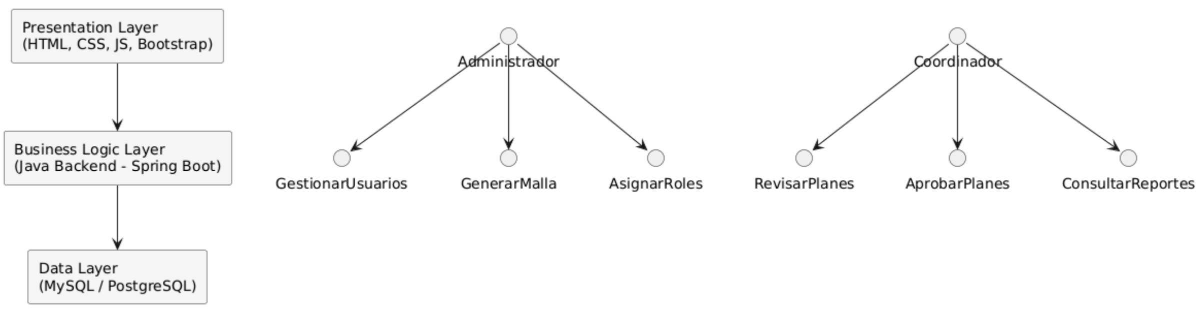


Information System Architecture

1. System Layers

1. **Presentation Layer (Frontend)**
    - Built with **HTML, CSS, JavaScript, Bootstrap**.
    - Teachers, coordinators, and administrators access it through their web browser.
    - Example screens:
        - Login.
        - Create/Edit subject area plan.
        - Generate curriculum map in PDF.
2. **Business Logic Layer (Backend in Java)**
    - Developed in **Java (Spring Boot or JSP/Servlets)**.
    - Responsible for:
        - Validating the data entered in forms.
        - Processing the creation and editing of subject area plans.
        - Generating PDF/Word documents.
        - Applying business rules (e.g., associating competencies with grade levels).
    - Exposes services (**REST APIs**) so the frontend can communicate with the database through the backend.
3. **Data Layer (Database)**
    - Database: **MySQL or PostgreSQL**.
    - Stores:
        - Users (teachers, administrators, coordinators).
        - Subject area plans.
        - Competencies and contents.
        - Generated curriculum maps.

General Scheme



3. Suggested Technologies

Frontend (Presentation Layer)

- **HTML5 / CSS3** → structure and design of the interfaces.
- **JavaScript (JS)** → basic interactivity.
- **Bootstrap** → responsive and user-friendly design.

Backend (Business Logic Layer)

- **Java** (with Spring Boot or basic Servlets) → application engine, business rules.
- **Maven or Gradle** → dependency management and build automation.
- **REST API** → communication between frontend and backend.

Database (Data Layer)

- **MySQL** or **PostgreSQL** → storage of users, subject plans, contents, and curriculum map.
- **JDBC** (for a basic approach) or **JPA/Hibernate** (for a more structured one).

Supporting Tools

- **GitHub / GitLab** → version control.
- **PlantUML** → architecture and use cases.
- **NetBeans / IntelliJ IDEA / Eclipse** → IDE for development.

Main Tables (Entities) and Attributes

User

- id_user (PK)
- name
- email
- password
- role (teacher, coordinator, administrator)

AreaPlan

- id_plan (PK)
- plan_name
- grade
- year
- id_user (FK → User)

Content

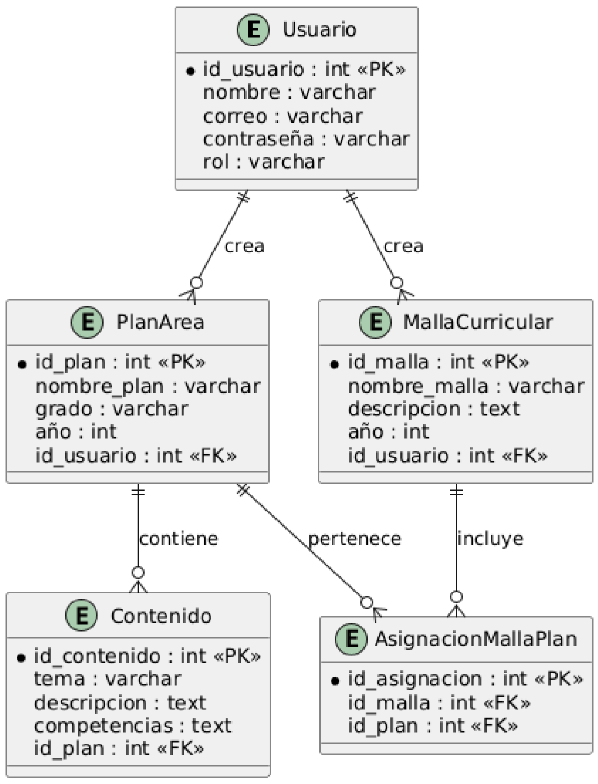
- id_content (PK)
- topic
- description
- competencies
- id_plan (FK → AreaPlan)

CurriculumGrid

- id_grid (PK)
- grid_name
- description
- year
- id_user (FK → User who creates it)

GridPlanAssignment (many-to-many relationship between CurriculumGrid and AreaPlan)

- id_assignment (PK)
- id_grid (FK → CurriculumGrid)
- id_plan (FK → AreaPlan)



Relational Model

CREATE TABLE Usuario (

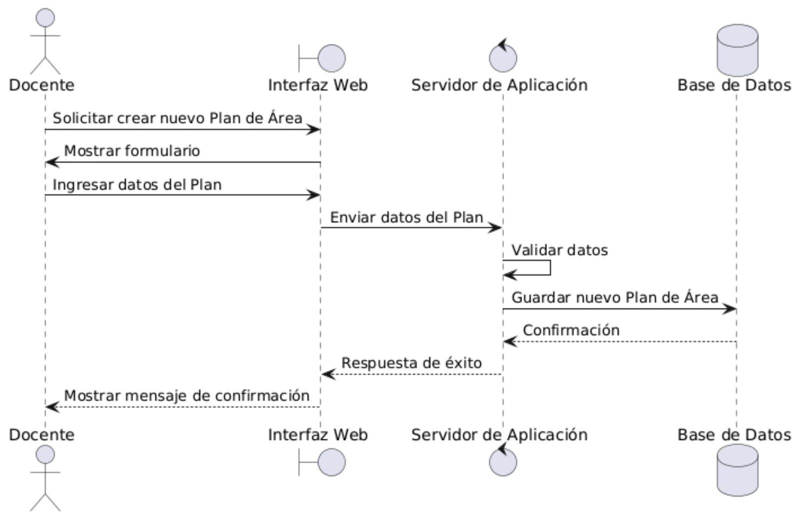
```
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,

    nombre VARCHAR(100) NOT NULL,

    correo VARCHAR(100) UNIQUE NOT NULL,

    contraseña VARCHAR(100) NOT NULL,

    rol ENUM('docente', 'coordinador', 'administrador') NOT NULL

);


CREATE TABLE PlanArea (

    id_plan INT PRIMARY KEY AUTO_INCREMENT,

    nombre_plan VARCHAR(100) NOT NULL,

    grado VARCHAR(50) NOT NULL,

    año INT NOT NULL,

    id_usuario INT,

    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)

);


CREATE TABLE Contenido (

    id_contenido INT PRIMARY KEY AUTO_INCREMENT,

    tema VARCHAR(100) NOT NULL,

    descripcion TEXT,

    competencias TEXT,

    id_plan INT,

    FOREIGN KEY (id_plan) REFERENCES PlanArea(id_plan)

);


CREATE TABLE MallaCurricular (

    id_malla INT PRIMARY KEY AUTO_INCREMENT,

    nombre_malla VARCHAR(100) NOT NULL,

    descripcion TEXT,

    año INT NOT NULL,

    id_usuario INT,
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)
);


CREATE TABLE AsignacionMallaPlan (
    id_asignacion INT PRIMARY KEY AUTO_INCREMENT,
    id_malla INT,
    id_plan INT,
    FOREIGN KEY (id_malla) REFERENCES MallaCurricular(id_malla),
    FOREIGN KEY (id_plan) REFERENCES PlanArea(id_plan)
);
```

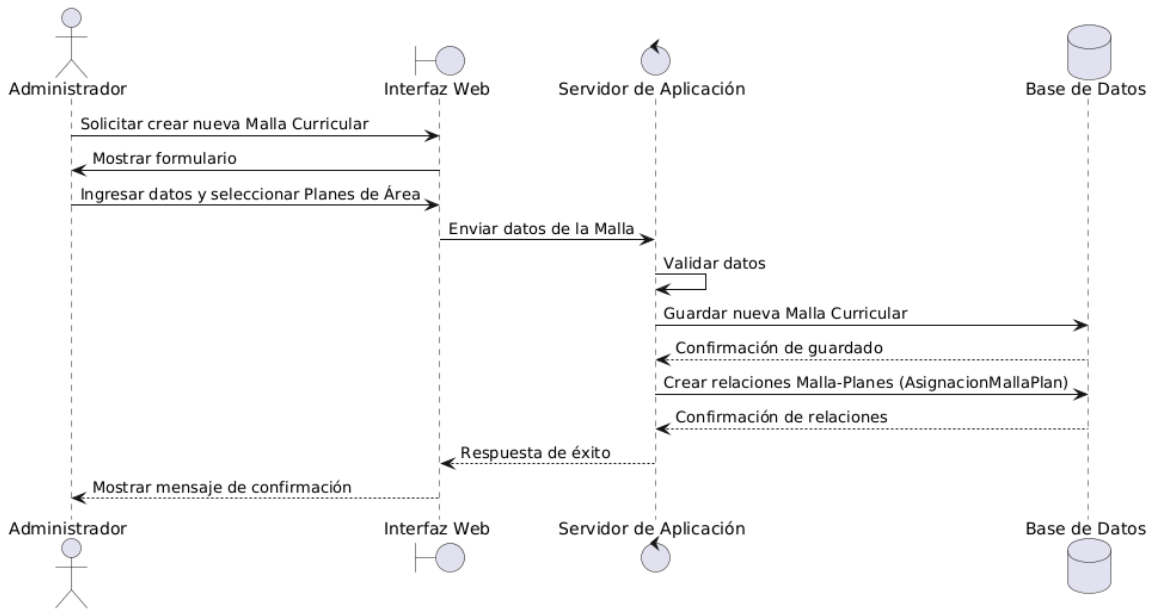
Physical Model

```
-- Crear base de datos
CREATE DATABASE IF NOT EXISTS MallaCurricularAdventista
    DEFAULT CHARACTER SET utf8mb4
    DEFAULT COLLATE utf8mb4_general_ci;


USE MallaCurricularAdventista;


-- Tabla de usuarios
CREATE TABLE Usuario (
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    correo VARCHAR(100) UNIQUE NOT NULL,
    contraseña VARCHAR(255) NOT NULL,
    rol ENUM('docente', 'coordinador', 'administrador') NOT NULL
) ENGINE=InnoDB;


-- Tabla de planes de área
CREATE TABLE PlanArea (
    id_plan INT AUTO_INCREMENT PRIMARY KEY,
```

```sql
    nombre_plan VARCHAR(150) NOT NULL,

    grado VARCHAR(50) NOT NULL,

    año YEAR NOT NULL,

    id_usuario INT NOT NULL,

    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)

        ON DELETE CASCADE

        ON UPDATE CASCADE

) ENGINE=InnoDB;


-- Tabla de contenidos

CREATE TABLE Contenido (

    id_contenido INT AUTO_INCREMENT PRIMARY KEY,

    tema VARCHAR(150) NOT NULL,

    descripcion TEXT,

    competencias TEXT,

    id_plan INT NOT NULL,

    FOREIGN KEY (id_plan) REFERENCES PlanArea(id_plan)

        ON DELETE CASCADE

        ON UPDATE CASCADE

) ENGINE=InnoDB;


-- Tabla de mallas curriculares

CREATE TABLE MallaCurricular (

    id_malla INT AUTO_INCREMENT PRIMARY KEY,

    nombre_malla VARCHAR(150) NOT NULL,

    descripcion TEXT,

    año YEAR NOT NULL,

    id_usuario INT NOT NULL,

    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)

        ON DELETE CASCADE

        ON UPDATE CASCADE
```

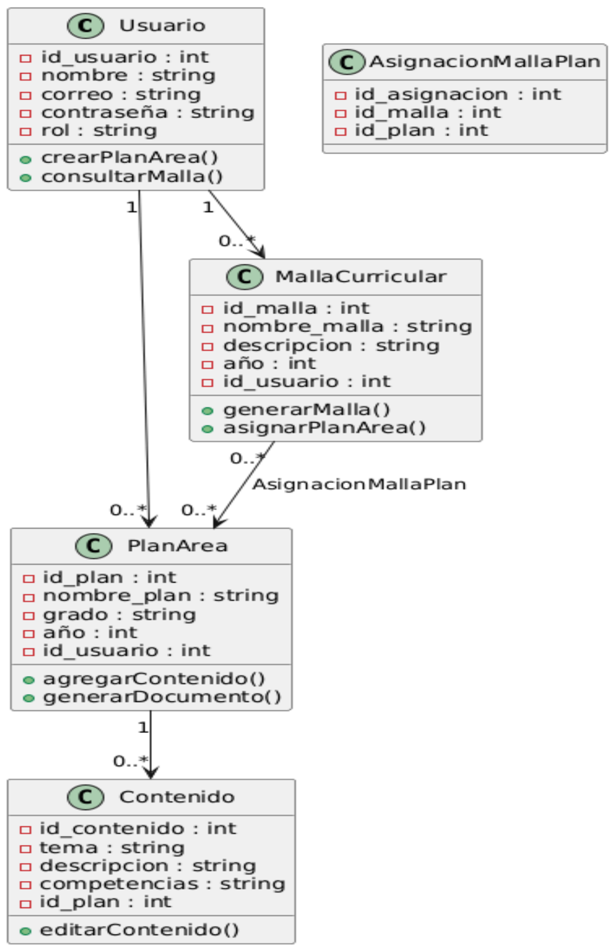
```
) ENGINE=InnoDB;


-- Tabla intermedia para relación muchos a muchos

CREATE TABLE AsignacionMallaPlan (

  id_asignacion INT AUTO_INCREMENT PRIMARY KEY,

  id_malla INT NOT NULL,

  id_plan INT NOT NULL,

  FOREIGN KEY (id_malla) REFERENCES MallaCurricular(id_malla)

    ON DELETE CASCADE

    ON UPDATE CASCADE,

  FOREIGN KEY (id_plan) REFERENCES PlanArea(id_plan)

    ON DELETE CASCADE

    ON UPDATE CASCADE

) ENGINE=InnoDB;
```



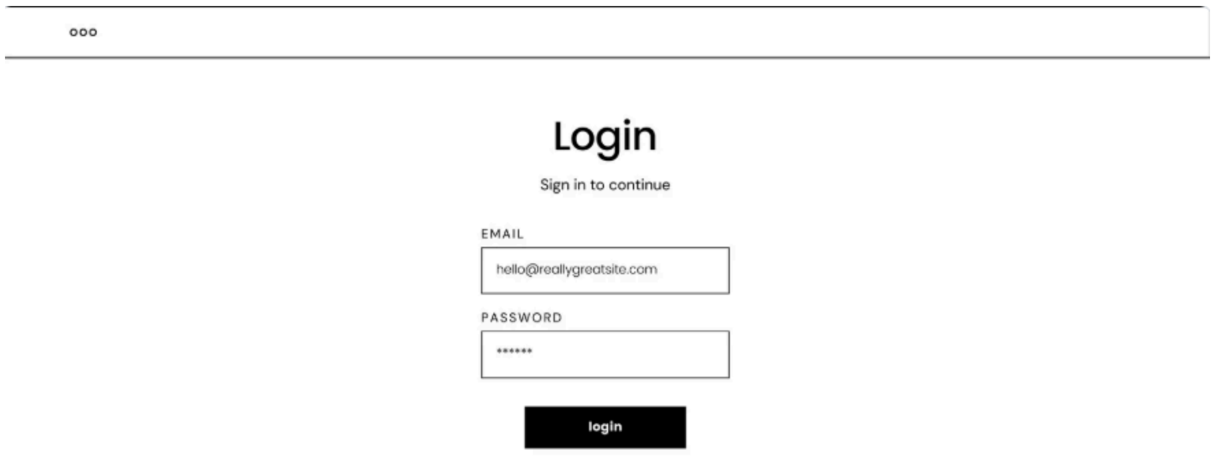


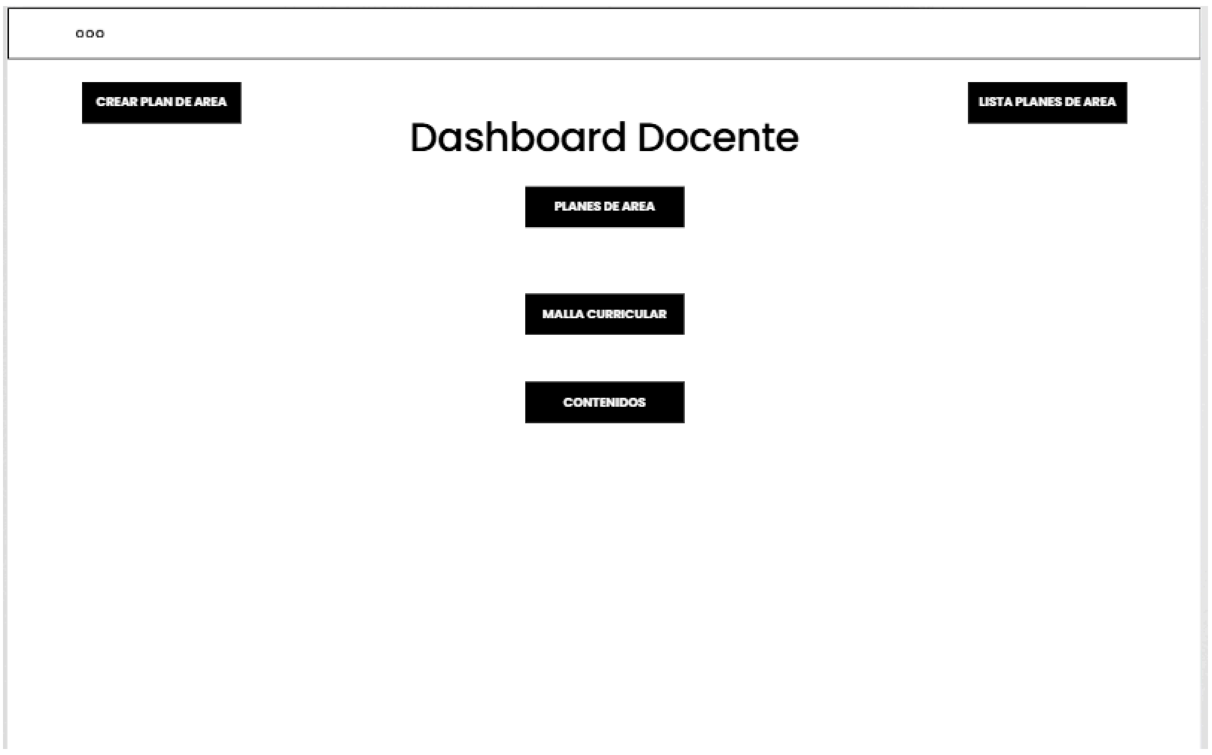
Administrador
Interfaz Web
Servidor de Aplicación
Base de Datos
Solicitar crear nueva Malla Curricular
Mostrar formulario
Ingresar datos y seleccionar Planes de Área
Enviar datos de la Malla
Validar datos
Guardar nueva Malla Curricular
Confirmación de guardado
Crear relaciones Malla-Planes (AsignacionMallaPlan)
Confirmación de relaciones
Respuesta de éxito
Mostrar mensaje de confirmación
Administrador
Interfaz Web
Servidor de Aplicación
Base de Datos

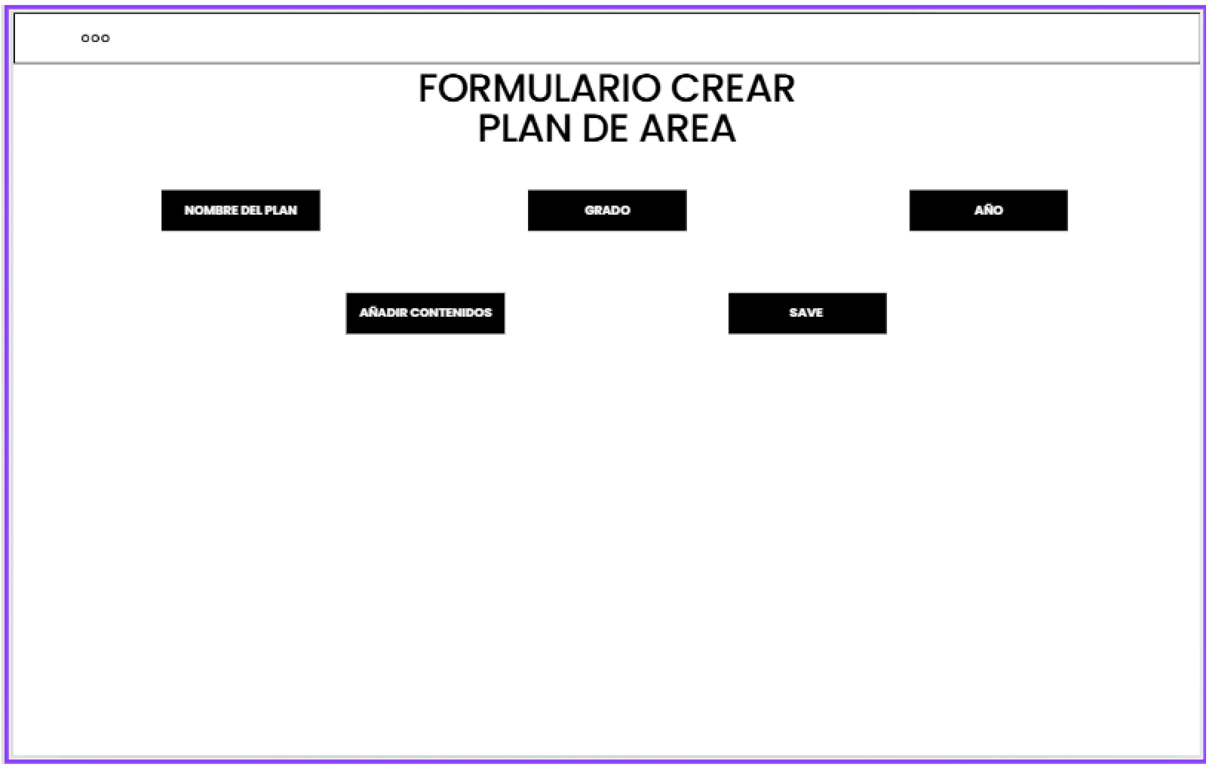
Diagrama de Clases de Uso

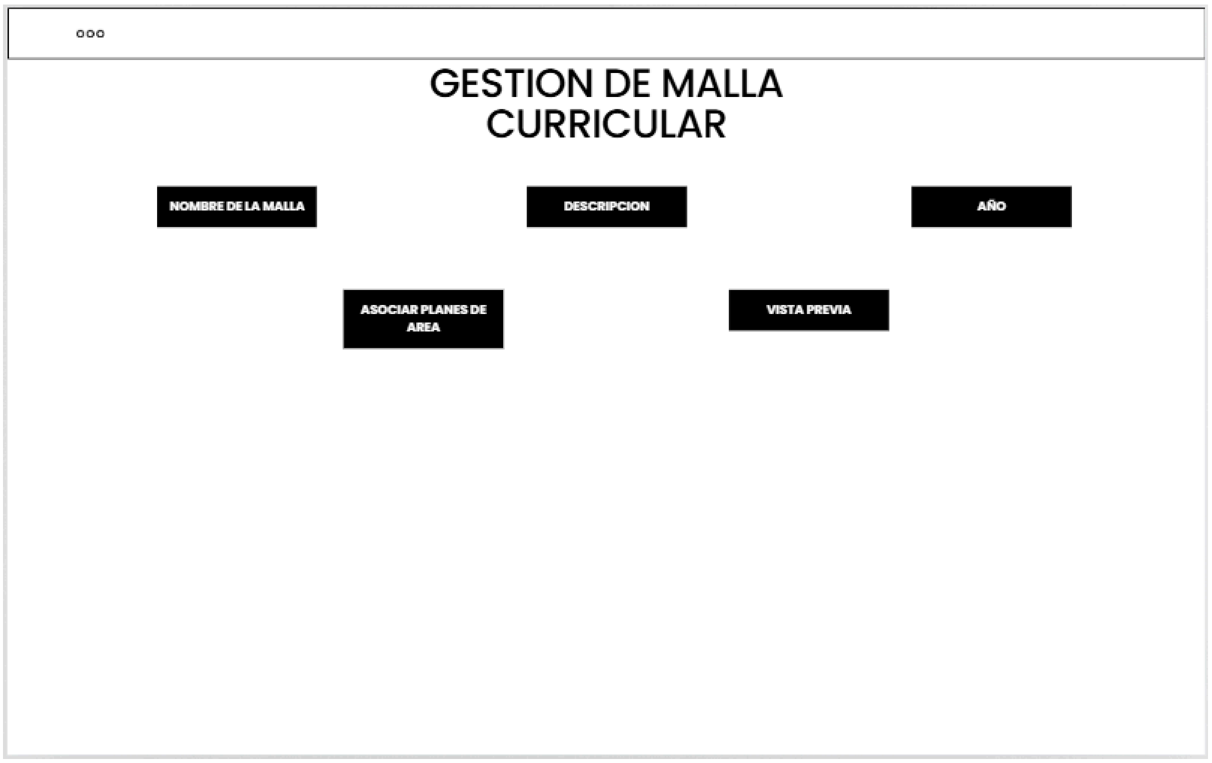


MOCKUPS


ooo
Login
Sign in to continue
EMAIL
hello@reallygreatsite.com
PASSWORD
******
login


ooo
CREAR PLAN DE AREA
LISTA PLANES DE AREA
Dashboard Docente
PLANES DE AREA
MALLA CURRICULAR
CONTENIDOS


FORMULARIO CREAR PLAN DE AREA
NOMBRE DEL PLAN
GRADO
AÑO
AÑADIR CONTENIDOS
SAVE


GESTION DE MALLA CURRICULAR
NOMBRE DE LA MALLA
DESCRIPCION
AÑO
ASOCIAR PLANES DE AREA
VISTA PREVIA

MVP – Religion Curriculum Platform

1. User Management

- User registration (teachers, coordinators, administrators).
- Login (email and password).
- Basic roles:
  - **Teacher** → creates area plans and contents.
  - **Administrator** → manages users and reviews curriculum grids.

2. Area Plan Management

- Create an area plan (name, grade, year).
- Associate contents with the plan (topic, description, competencies).
- Save and list created plans.

3. Curriculum Grid Management

- Create a curriculum grid (name, description, year).
- Associate area plans with the grid.
- View the curriculum grid.

4. Basic Export

- Generate a **PDF document** of the area plan and curriculum grid.

5. Minimal Interface

- **Login / Registration.**
- **Teacher Dashboard** with quick access to:
  - Area plans.

    - Curriculum grids.

- Simple forms to create plans and grids.

Sprint Planning – Curriculum Grid Platform

Sprint 1 – Users and Authentication

- User registration (teacher, coordinator, administrator).
- Login with email and password.
- Assignment of basic roles.
  Objective: Have the authentication module and basic user management ready.

Sprint 2 – Plans and Curriculum Management

- Create Area Plan (name, grade, year).
- Associate contents to the plans (topic, description, competencies).
- Create Curriculum Grid and associate Area Plans.
- View created grids and plans.
  Objective: Allow the basic construction of area plans and their integration into a curriculum grid.

Sprint 3 – MVP Consolidation

- Export Area Plan to PDF.
- Export Curriculum Grid to PDF.
- Basic dashboard for teachers and administrators.
- Interface improvements (clear forms, simple navigation).
- Administration functions (edit/delete plans, grids, and users).
  Objective: Consolidate the MVP with export, usability, and basic administration.

User Manual – Religion Curriculum App

Introduction

The **Religion Curriculum App** is a desktop application built with **JavaFX**. It helps teachers and administrators manage, edit, and approve curriculum plans for the subject of Religion. The app provides a simple interface for creating and organizing area plans.

Login

1. Start the application.
2. The **Login Screen** will appear.
3. Enter your username and password.
4. Click **Login** to access the dashboard.

If your credentials are correct, the system will open the main dashboard.

Dashboard Overview

After logging in, you will see the **Dashboard**, divided into two modules:

1. **Area Plans (Planes de Área)** – main module to manage curriculum plans.
2. **Users (Usuarios)** – currently disabled, reserved for future user administration.

Managing Area Plans

Inside the **Area Plans tab**, you can:

- **Search Plans**: Use the search box to find plans by title, level, or grade.
- **Create a New Plan**:

    - Click **New**.
    - Fill in the form fields: title, level, grade, contents, competencies, and state.
    - Click **Save** to store the plan.
- **Edit a Plan**:
    - Select a plan from the table.
    - Click **Edit** to open it in the form.
    - Modify the information and save changes.
- **Delete a Plan**:
    - Select a plan from the table.
    - Click **Delete** to remove it.
- **Approve a Plan**:
    - Select a plan from the table.
    - Click **Approve** to mark it as approved.

Fields in the Plan Form

- **Title** – Name of the plan (e.g., "Annual Religion Plan").
- **Level** – Select the education level (e.g., Elementary, Secondary).
- **Grade** – Select the grade related to the plan.
- **Contents** – Specify themes, units, or topics.
- **Competencies** – Define objectives, skills, and expected outcomes.
- **State** – Choose the current status (Draft, Approved, etc.).

Additional Notes

- The **Users tab** is not active yet. It will be available in future updates.
- All data is displayed in a table for easy access and management.
- Plans can only be edited or approved if first selected in the list.