



Ingeniería de Software: Sistema de Convalidaciones Curriculares

Juan Esteban Graell Alzate

Universidad Nacional de Colombia
Facultad de Administración, Departamento de Informática y Computación
Manizales, Caldas
2025

Índice

Introducción.....	2
Contexto y Justificación.....	2
Objetivos generales y específicos.....	2
Objetivo general.....	2
Objetivos específicos.....	2
Alcance del Sistema.....	3
Funcionalidades Incluidas.....	3
Límites y Exclusiones.....	4
Requisitos del sistema.....	5
Requisitos funcionales.....	5
Requisitos no funcionales.....	6
Casos de uso.....	7
Bloque 1. Importación y listados.....	7
Bloque 2. Gestión de convalidaciones.....	7
Bloque 3. Simulación curricular.....	8
Resumen de implementación.....	8
Actores del sistema.....	8
Trazabilidad con requisitos funcionales.....	9
Diagramas de casos de uso.....	9
Arquitectura del sistema.....	12
Modelo de datos.....	13
Diseño de la interfaz de usuario.....	14

Introducción

Contexto y Justificación

La gestión académica universitaria enfrenta el reto constante de garantizar la equivalencia y coherencia entre planes de estudio, especialmente en escenarios de actualización curricular o procesos de convalidación de asignaturas. Estos procedimientos suelen ser complejos, demandando tiempo y esfuerzo tanto por parte de coordinadores académicos como de estudiantes, quienes requieren herramientas que faciliten la toma de decisiones y reduzcan los errores asociados a la validación manual de contenidos.

En el contexto del programa de Administración de Sistemas Informáticos de la Universidad Nacional de Colombia, los procesos de convalidación involucran:

- Evaluación manual de equivalencias entre materias de diferentes instituciones
- Análisis complejo de prerrequisitos y dependencias curriculares
- Cálculo de porcentajes de progreso académico tras migraciones curriculares
- Documentación dispersa y procesos poco estandarizados
- Tiempo excesivo para resolución de casos de convalidación

Objetivos generales y específicos

Objetivo general

Desarrollar un sistema web integral de convalidaciones curriculares que automatice el mapeo, análisis y gestión de equivalencias entre mallas curriculares externas o nuevas y la malla base del programa de Administración de Sistemas Informáticos, con el fin de optimizar los tiempos de procesamiento y garantizar la consistencia en los criterios de convalidación.

Objetivos específicos

1. Sistema de importación automatizada

- Implementar carga automatizada de currículos externos o nuevos desde archivos CSV/Excel.

2. Motor de convalidaciones inteligente

- Desarrollar sistema de mapeo manual entre materias externas o nuevas e internas.
- Implementar tres tipos de convalidación: directa, libre elección y no convalidada.

3. Herramientas de simulación curricular

- Crear interfaz interactiva de visualización de malla curricular con funcionalidad drag-and-drop.
- Implementar validación automática de prerrequisitos durante simulaciones.
- Desarrollar análisis de impacto en tiempo real para cambios curriculares.

4. Sistema de análisis y métricas

- Calcular automáticamente porcentajes de convalidación y progreso académico.
- Generar métricas de impacto sobre poblaciones estudiantiles.
- Proporcionar dashboard de estadísticas en tiempo real.

5. Arquitectura escalable y mantenible

- Implementar arquitectura MVC con Laravel para facilitar el mantenimiento.
- Configurar despliegue containerizado para replicabilidad.
- Establecer estructura de datos normalizada y extensible.

Alcance del Sistema

Funcionalidades Incluidas

1. Módulo de Convalidaciones:

- Gestión completa de currículos externos o nuevos (CRUD)
- Sistema de convalidaciones con tres modalidades: directa (con porcentaje de equivalencia), libre elección (créditos electivos), y no convalidada (requisitos adicionales)
- Dashboard estadístico con métricas de progreso y completitud

2. Módulo de Simulación Curricular:

- Visualización interactiva de malla curricular organizada por semestres
- Funcionalidad drag-and-drop para reorganización temporal de materias
- Validación automática de prerrequisitos y dependencias
- Análisis de impacto en tiempo real sobre estudiantes activos
- Exportación de configuraciones curriculares modificadas

3. Módulo de Gestión de Datos:

- Importación de datos académicos históricos desde CSV
- Gestión de información de estudiantes para análisis de impacto

4. Infraestructura Técnica:

- Aplicación web desarrollada en Laravel 12 con PHP 8.2+
- Base de datos PostgreSQL 15 con esquema optimizado
- Interfaz responsive con JavaScript vanilla y CSS personalizado
- Despliegue containerizado con Docker Compose (Nginx + PHP-FPM + PostgreSQL)
- Scripts automatizados de configuración y mantenimiento

Límites y Exclusiones

No incluye:

- Sistema de autenticación y autorización por roles
- Integración con sistemas académicos institucionales (SIA, plataformas LMS)
- Generación de reportes oficiales en PDF para trámites administrativos
- Gestión de múltiples programas académicos simultáneos
- Sistema de notificaciones automáticas a usuarios
- Workflow de aprobación institucional para convalidaciones

- Auditoría completa de cambios con trazabilidad histórica
- Funcionalidades de calificaciones y cálculo de promedios ponderados

Contexto de uso:

- Diseñado para uso interno del Departamento de Informática y Computación
- Operación local/desarrollo sin requisitos de alta disponibilidad
- Enfoque específico en el programa de Administración de Sistemas Informáticos
- Uso por parte de coordinadores académicos y personal administrativo especializado

Requisitos del sistema

Requisitos funcionales

Los siguientes requisitos funcionales (RF) están alineados con la funcionalidad implementada en el proyecto y verificables en el código fuente (rutas, controladores, modelos y vistas):

1. **RF-01. Carga de mallas curriculares externas y nuevas (CSV).** El sistema debe permitir la carga de mallas externas y nuevas desde archivos CSV válidos y almacenar sus materias asociadas.
2. **RF-02. Visualización de malla interna por semestres.** El sistema debe mostrar la malla interna organizada por semestre con sus materias y relaciones de prerrequisito.
3. **RF-03. Gestión de convalidaciones por materia externa o nueva.** Para cada materia externa o nueva se debe poder crear o eliminar una convalidación con una materia interna, o marcarla como libre elección/no convalidada.
4. **RF-04. Tipos de convalidación.** El sistema debe soportar los tipos de convalidación: directa, libre elección y no convalidada.
5. **RF-05. Resumen y estadísticas por currículo externo.** El sistema debe mostrar totales y porcentajes de convalidación.
6. **RF-06. Exportación de reporte de convalidaciones (CSV/JSON).** El sistema debe permitir exportar el estado de convalidaciones en formato descargable.

7. **RF-07. Simulación curricular con drag-and-drop.** El sistema debe permitir reorganizar materias entre semestres de forma temporal.
8. **RF-08. Análisis de impacto de cambios.** El sistema debe calcular y mostrar el impacto de cambios curriculares sobre estudiantes actuales.
9. **RF-09. Importación de historial académico de estudiantes (CSV).** El sistema debe permitir cargar registros históricos de estudiantes.
10. **RF-10. Orden original y guardado de malla modificada.** El sistema debe permitir consultar el orden original y guardar una versión modificada de la malla con fines de análisis.

Requisitos no funcionales

1. **RNF-01. Rendimiento.** La vista principal debe cargar en ≤ 2 segundos en entorno local con hasta 100 materias; el análisis de impacto debe responder en ≤ 2 segundos con datasets de tamaño moderado.
2. **RNF-02. Usabilidad.** La interfaz debe ser intuitiva, consistente y ofrecer retroalimentación clara al usuario.
3. **RNF-03. Portabilidad.** El sistema debe desplegarse de forma reproducible mediante Docker Compose.
4. **RNF-04. Compatibilidad.** El sistema debe ser compatible con navegadores modernos e importar archivos CSV estándar (UTF-8).
5. **RNF-05. Seguridad básica.** Debe aplicar validaciones de entradas, sanitización de datos y protección CSRF, aunque no contempla autenticación de usuarios.
6. **RNF-06. Confiabilidad e integridad.** Debe garantizar integridad referencial en base de datos y manejar adecuadamente errores en importaciones.
7. **RNF-07. Mantenibilidad.** El sistema debe estructurarse bajo el patrón MVC con convenciones de código y pruebas básicas.
8. **RNF-08. Observabilidad.** Debe registrar eventos clave y permitir la inspección de logs en el contenedor.
9. **RNF-09. Escalabilidad base.** El modelo de datos debe permitir crecer a múltiples mallas y cientos de materias sin degradación significativa en entornos de prueba.

Casos de uso

En esta sección se presentan los principales casos de uso del Sistema de Convalidaciones Curriculares. Se organizan en tres bloques funcionales: (a) importación y listados, (b) gestión de convalidaciones y (c) simulación curricular. Para cada caso de uso se especifica el actor involucrado, la descripción funcional y el estado de implementación.

Bloque 1. Importación y listados

1. **CU-01.** Cargar malla curricular externa (CSV).
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Importar materias de una malla externa desde archivo CSV.
 - c. **Estado:** Implementado (Web).
2. **CU-02.** Listar mallas curriculares externas.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Visualizar catálogo de todas las mallas externas cargadas.
 - c. **Estado:** Implementado (Web).
3. **CU-03.** Visualizar malla interna por semestres.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Explorar la malla interna organizada por semestres (1–10).
 - c. **Estado:** Implementado (Web).
4. **CU-13.** Importar historial académico de estudiantes (CSV).
 - a. **Actor:** Coordinador académico / Administrador del sistema.
 - b. **Descripción:** Cargar registros históricos de estudiantes para análisis.
 - c. **Estado:** Implementado (CLI).
 - d. **Nota:** Disponible sólo vía línea de comandos por seguridad.

Bloque 2. Gestión de convalidaciones

5. **CU-04.** Crear convalidación para materia externa.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Establecer equivalencia entre materia externa e interna (directa, libre elección, no convalidada).
 - c. **Estado:** Implementado (Web).
6. **CU-05.** Eliminar convalidación existente.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Quitar equivalencia previamente creada.
 - c. **Estado:** Implementado (Web).
7. **CU-06.** Cambiar tipo de convalidación.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Ajustar el tipo de convalidación (directa/libre/no convalidada).
 - c. **Estado:** Implementado (Web).
8. **CU-07.** Obtener sugerencias de equivalencia.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Asistir mapeo mediante similitud de nombres.
 - c. **Estado:** Implementado (Web).

9. **CU-08.** Ver resumen y estadísticas de una malla externa.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Consultar avance y distribución por tipo de convalidación.
 - c. **Estado:** Implementado (Web).
10. **CU-09.** Exportar reporte de convalidaciones (CSV).
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Descargar estado de convalidaciones en formato CSV.
 - c. **Estado:** Implementado (Web).

Bloque 3. Simulación curricular

11. **CU-10.** Simular reorganización curricular (drag-and-drop).
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Reorganizar materias entre semestres para evaluar escenarios.
 - c. **Estado:** Implementado (Web).
12. **CU-11.** Validar prerequisites en simulación.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Prevenir configuraciones que violen dependencias curriculares.
 - c. **Estado:** Implementado (Web).
13. **CU-12.** Analizar el impacto de cambios curriculares.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Estimar porcentaje de estudiantes afectados por modificaciones.
 - c. **Estado:** Implementado (Web).
14. **CU-14.** Consultar el orden original de la malla.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Recuperar el orden base de la malla curricular.
 - c. **Estado:** Implementado (Web).
15. **CU-15.** Guardar malla modificada desde simulación.
 - a. **Actor:** Coordinador académico.
 - b. **Descripción:** Persistir una versión modificada tras la simulación.
 - c. **Estado:** Implementado (Web).

Resumen de implementación

- Total de casos de uso: 15
- Implementados (Web): 14 (93 %)
- Implementados (CLI): 1 (7 %) – CU-13
- Sin implementar: 0 (0 %)

Actores del sistema

- Coordinador académico (actor principal, usuario web)
- Administrador del sistema (operaciones CLI)
- Sistema de importación (servicio interno)
- Motor de simulación (validador de prerequisites)
- Base de datos (persistencia y cálculos)

Trazabilidad con requisitos funcionales

RF-01 : CU-01, CU-02

RF-02 : CU-03

RF-03 : CU-04, CU-05

RF-04 : CU-06

RF-05 : CU-07

RF-06 : CU-08

RF-07 : CU-09

RF-08 : CU-10

RF-09 : CU-11

RF-10 : CU-12

RF-11 : CU-13

RF-12 : CU-14, CU-15

Diagramas de casos de uso

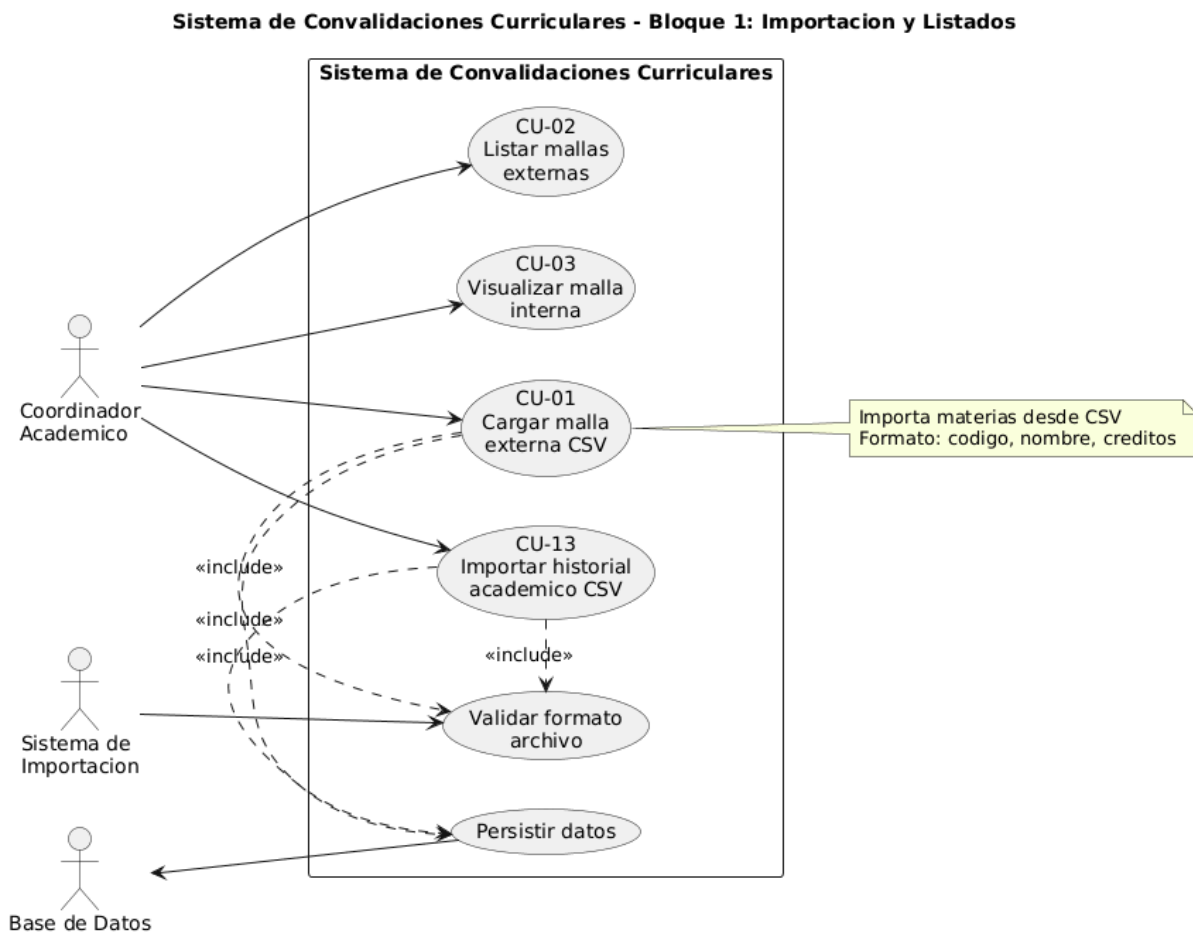


Figura 1. Diagrama de casos de uso del bloque 1

Sistema de Convalidaciones Curriculares - Bloque 2: Gestion de Convalidaciones

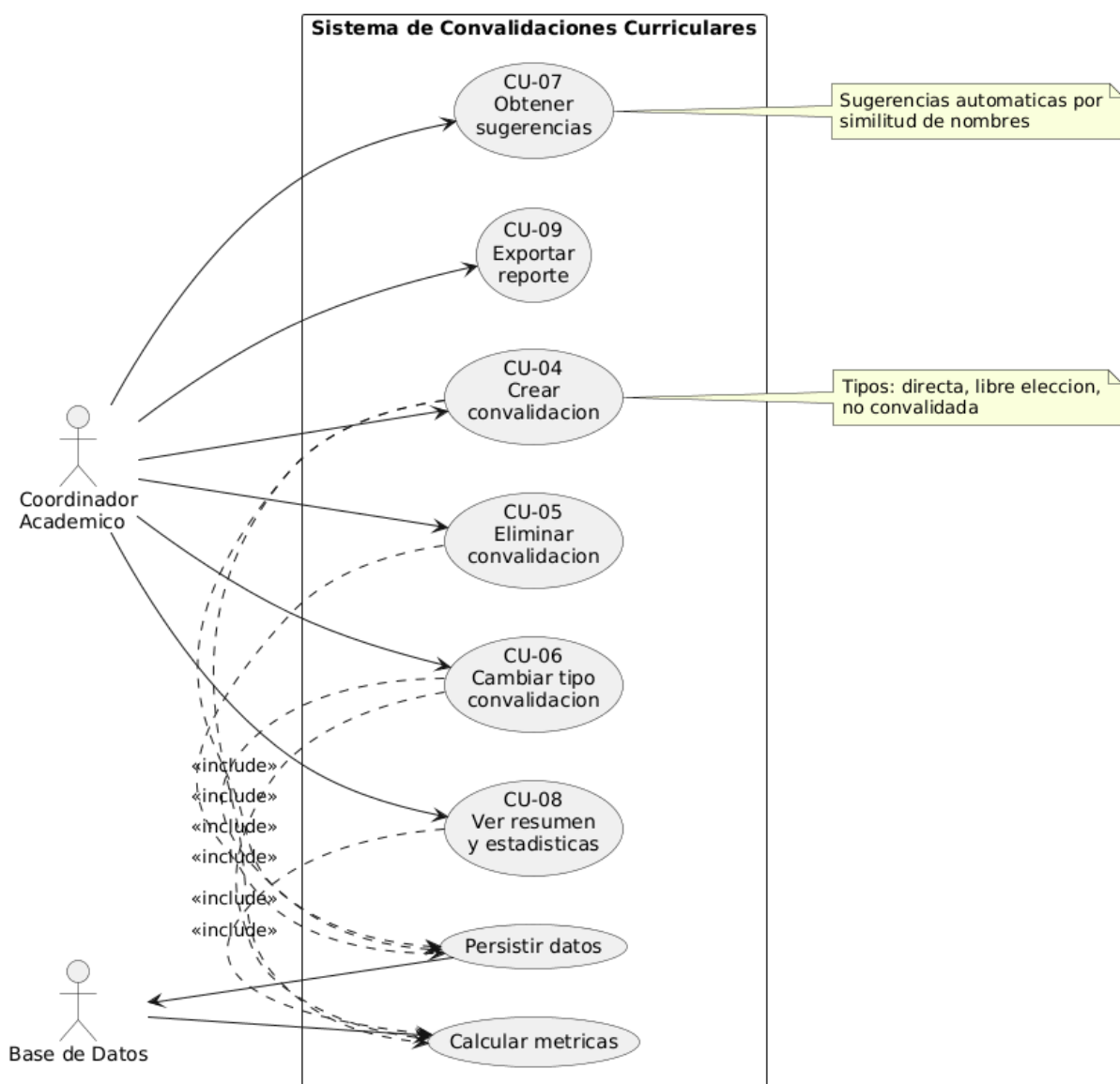


Figura 2. Diagrama de casos de uso del bloque 2

Sistema de Convalidaciones Curriculares - Bloque 3: Simulacion Curricular

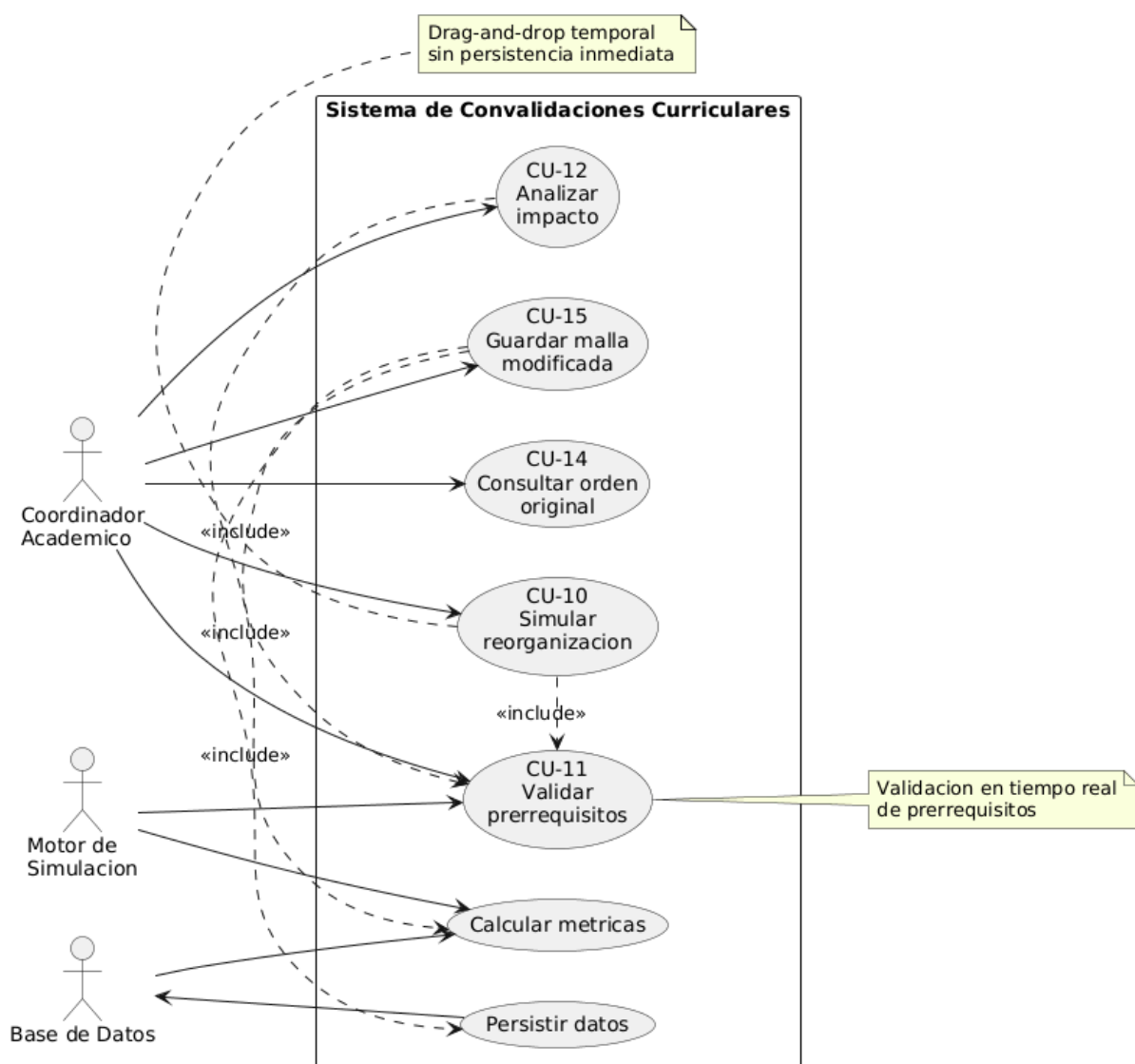
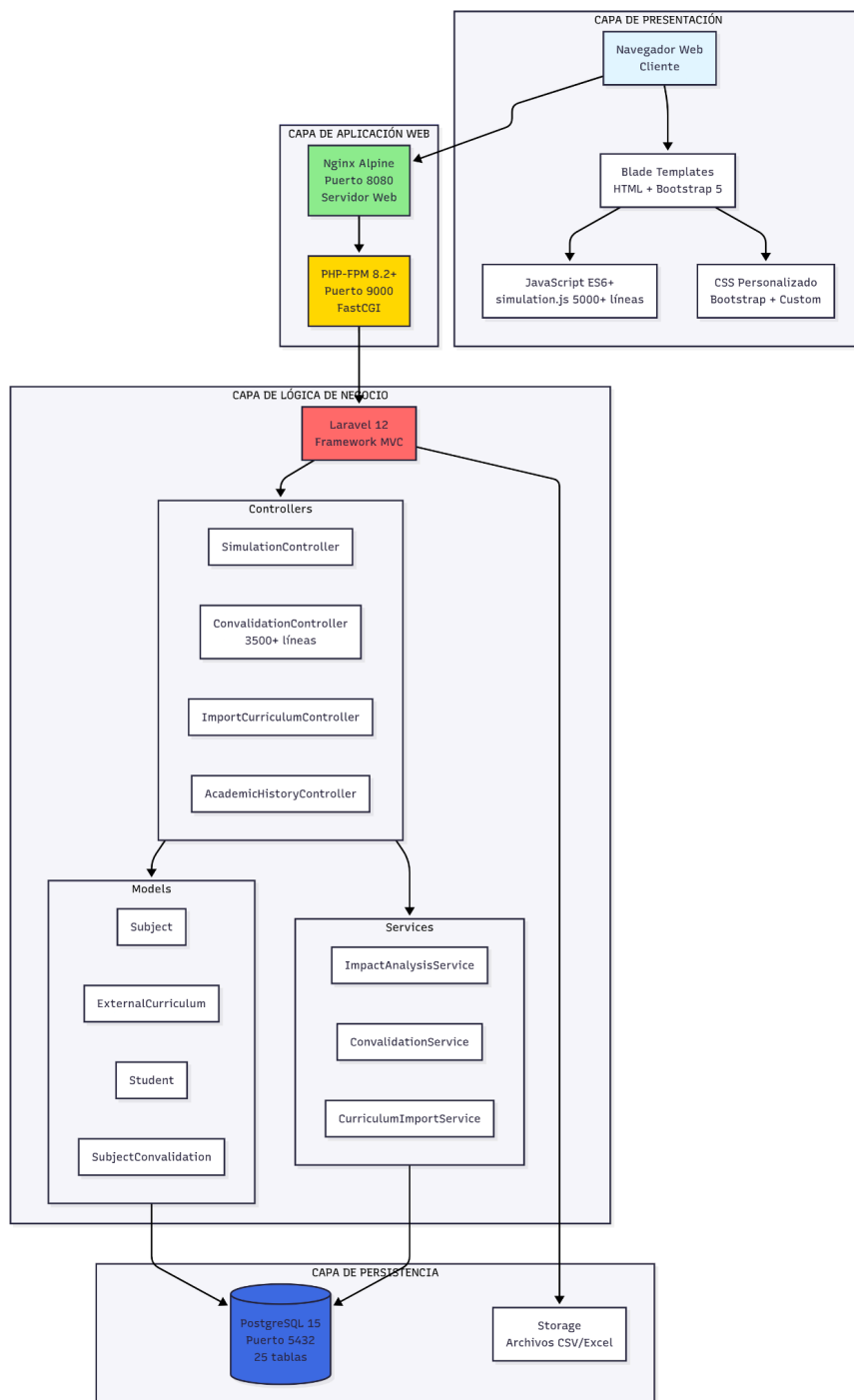


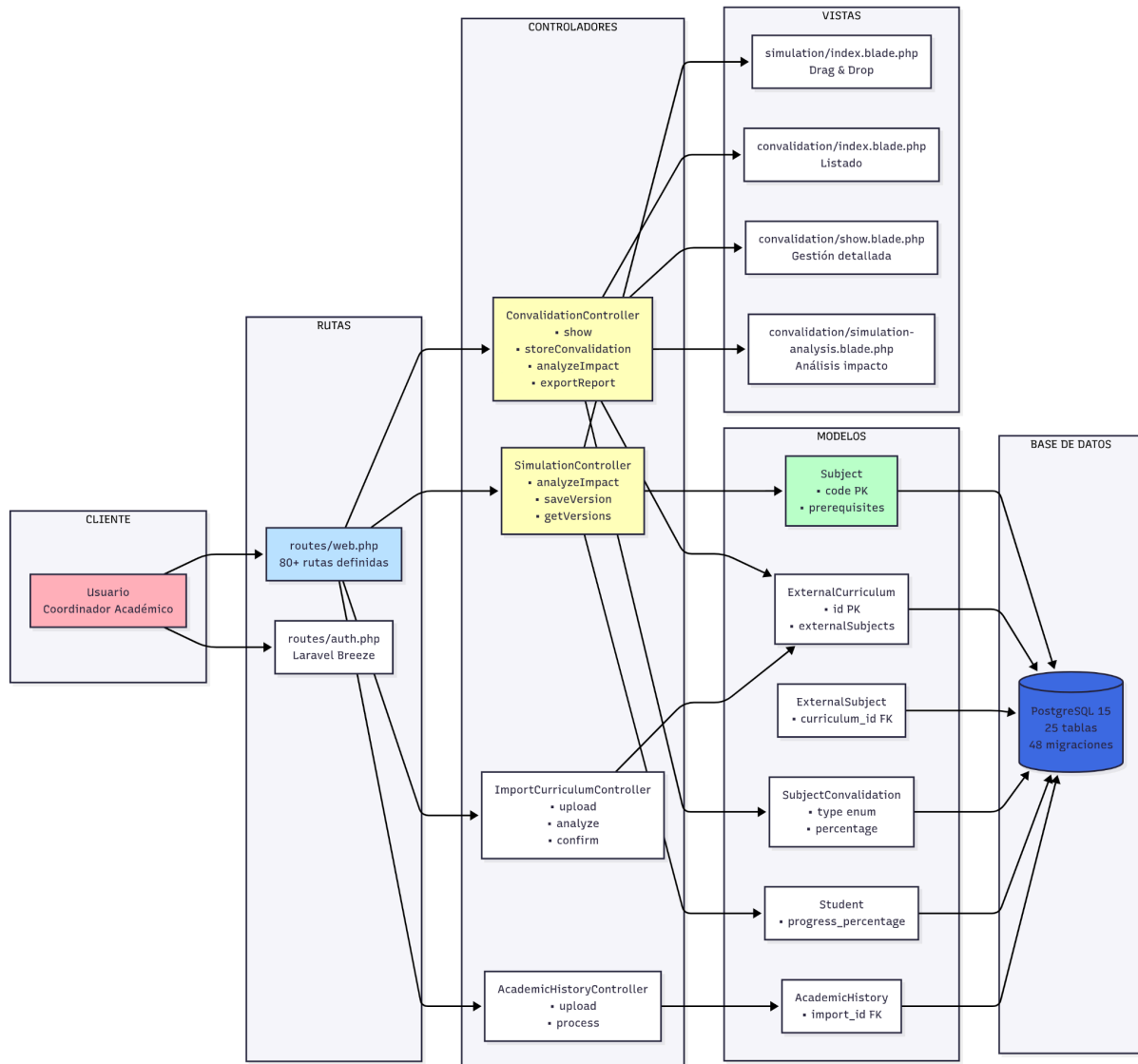
Figura 3. Diagrama de casos de uso del bloque 3

Arquitectura del sistema

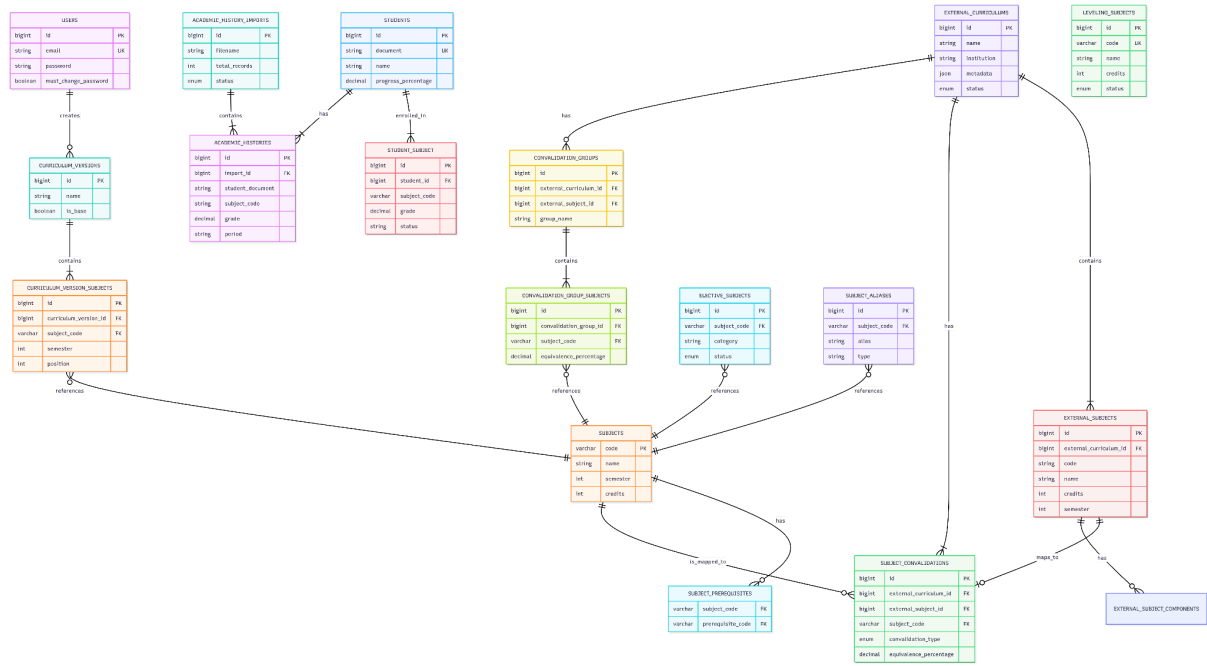
Arquitectura General (Vista de Capas)



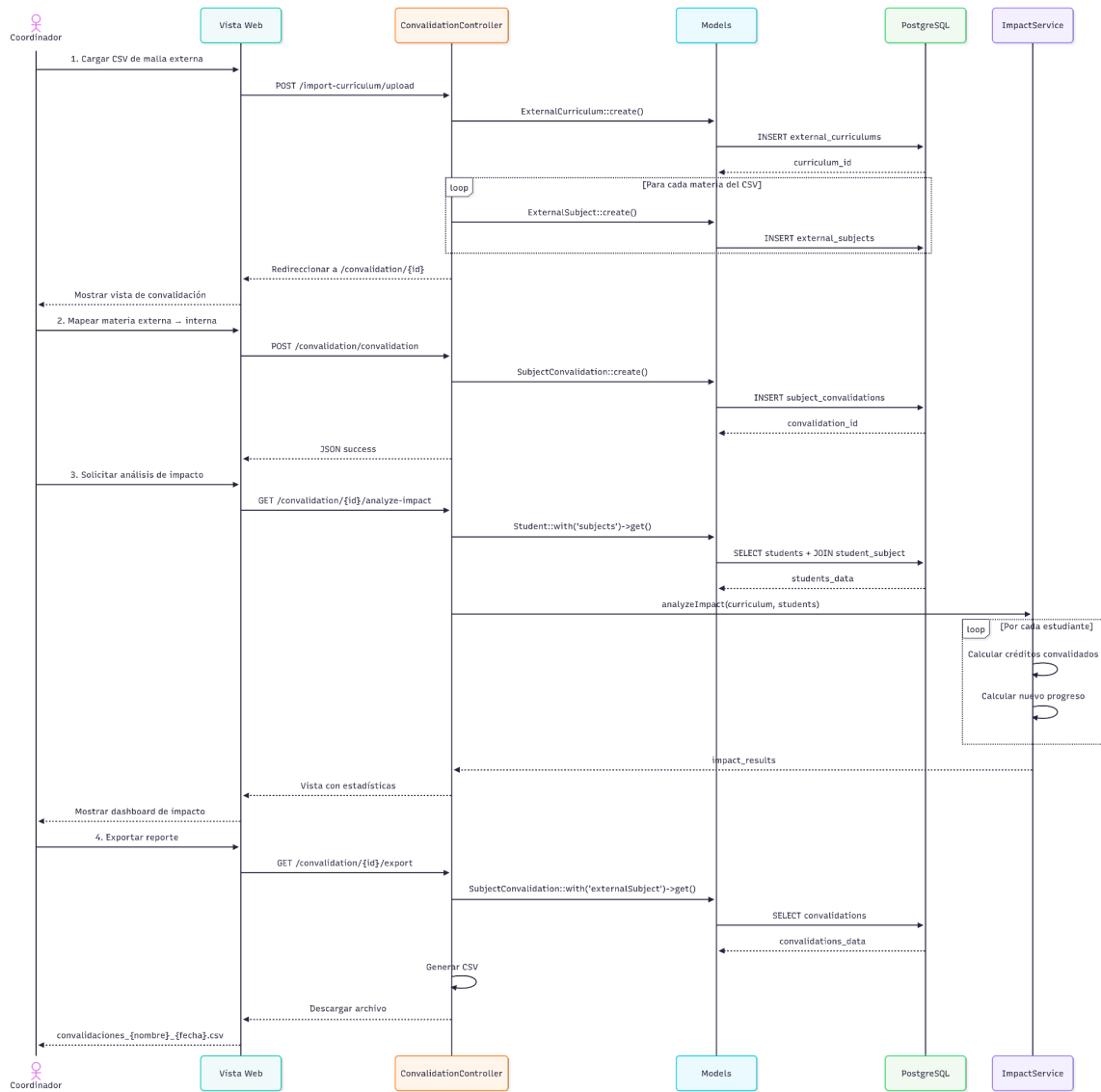
Arquitectura MVC de Laravel



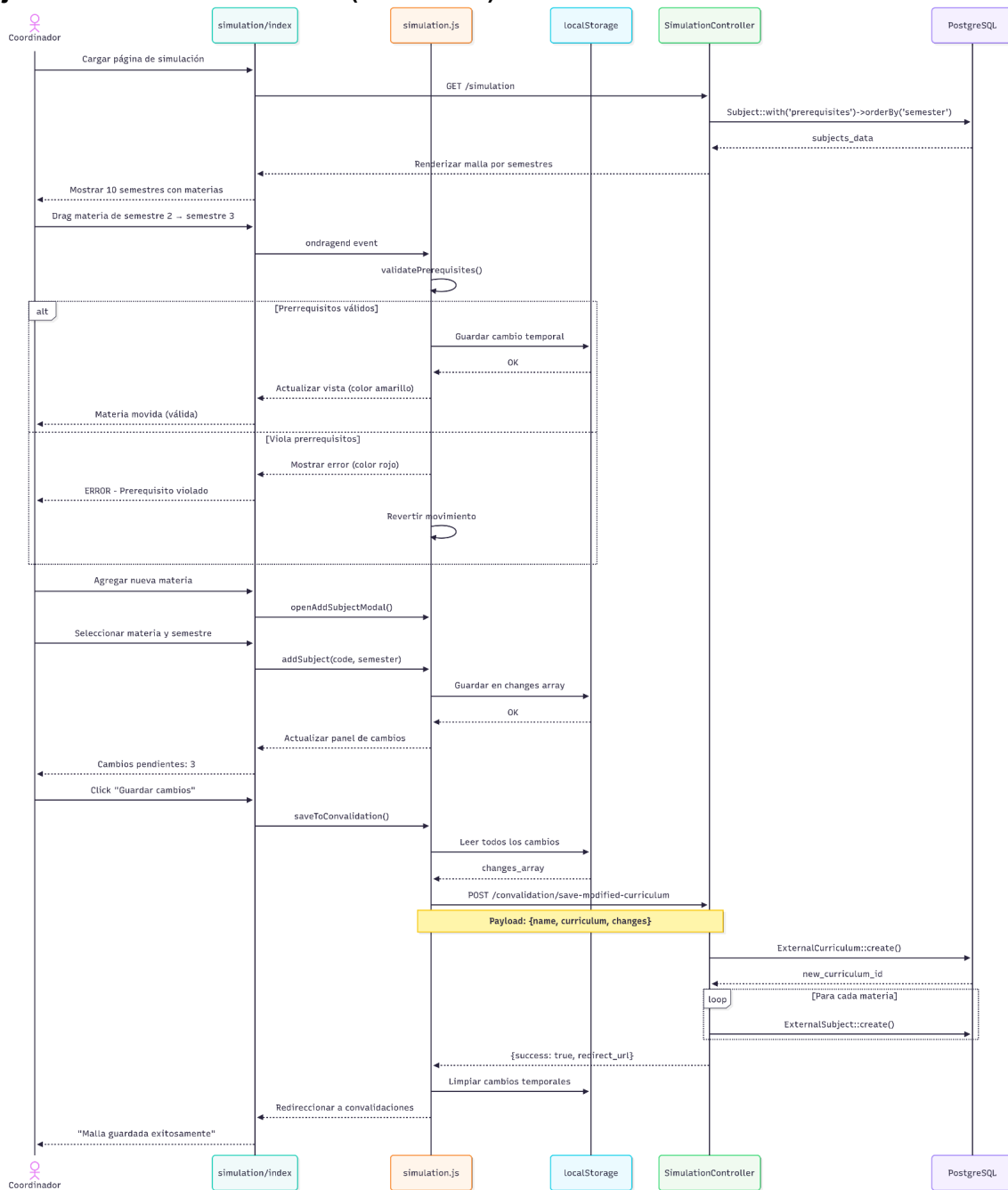
Modelo de Datos (Entidad- Relación Simplificado)



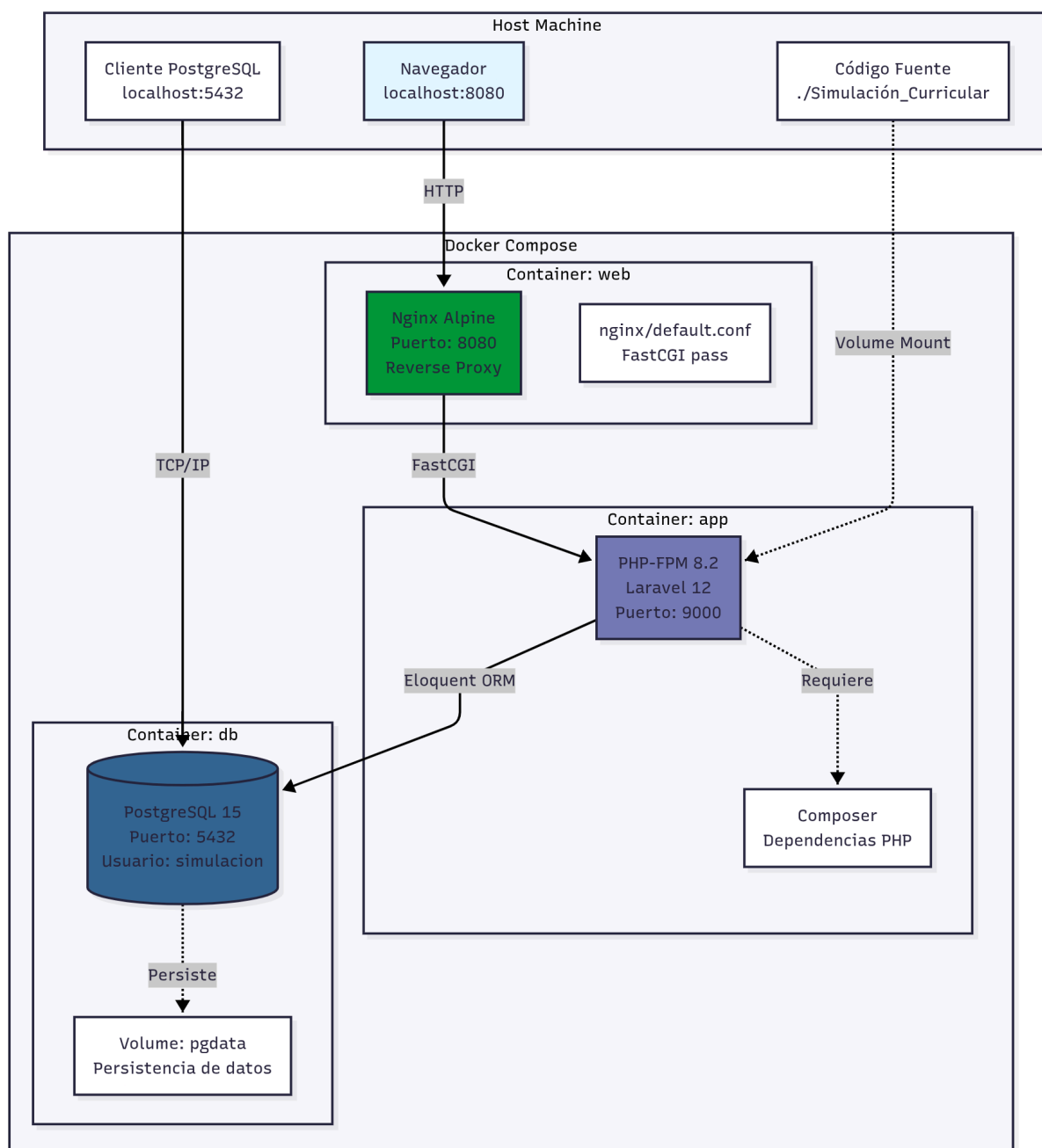
Flujo de Convalidación (Secuencia)



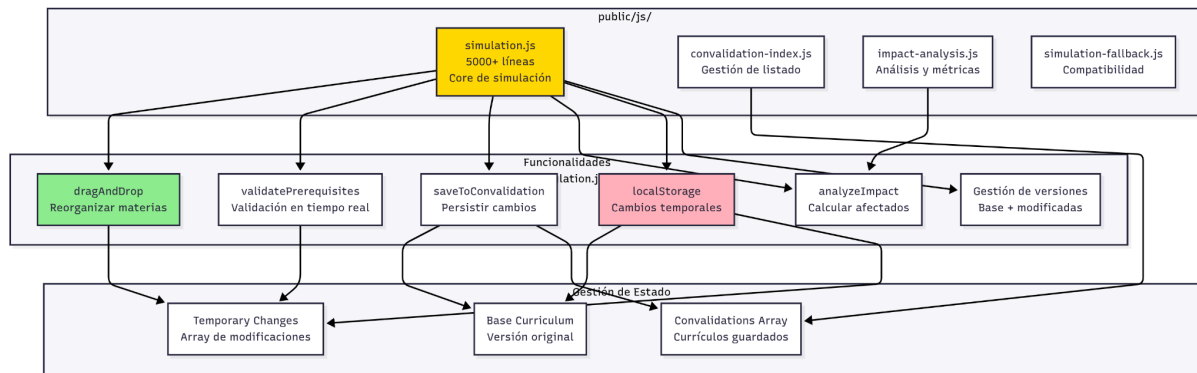
Flujo de Simulación Curricular (Secuencia)



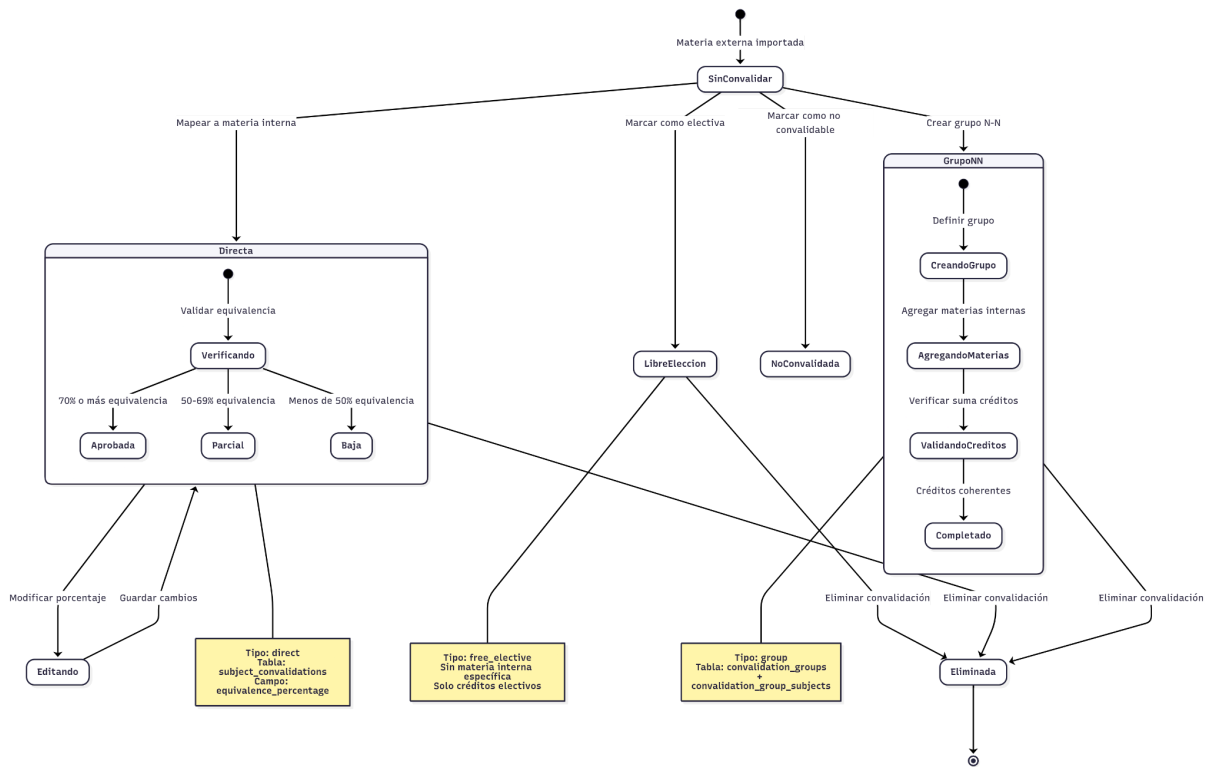
Infraestructura Docker (Contenedores)



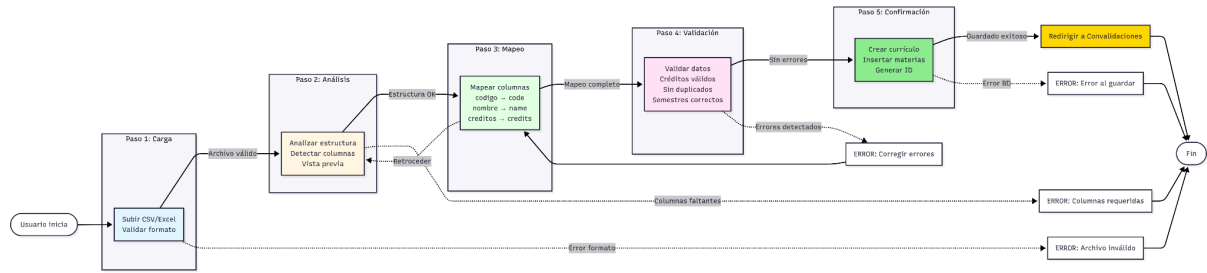
Componentes Frontend (JavaScript)



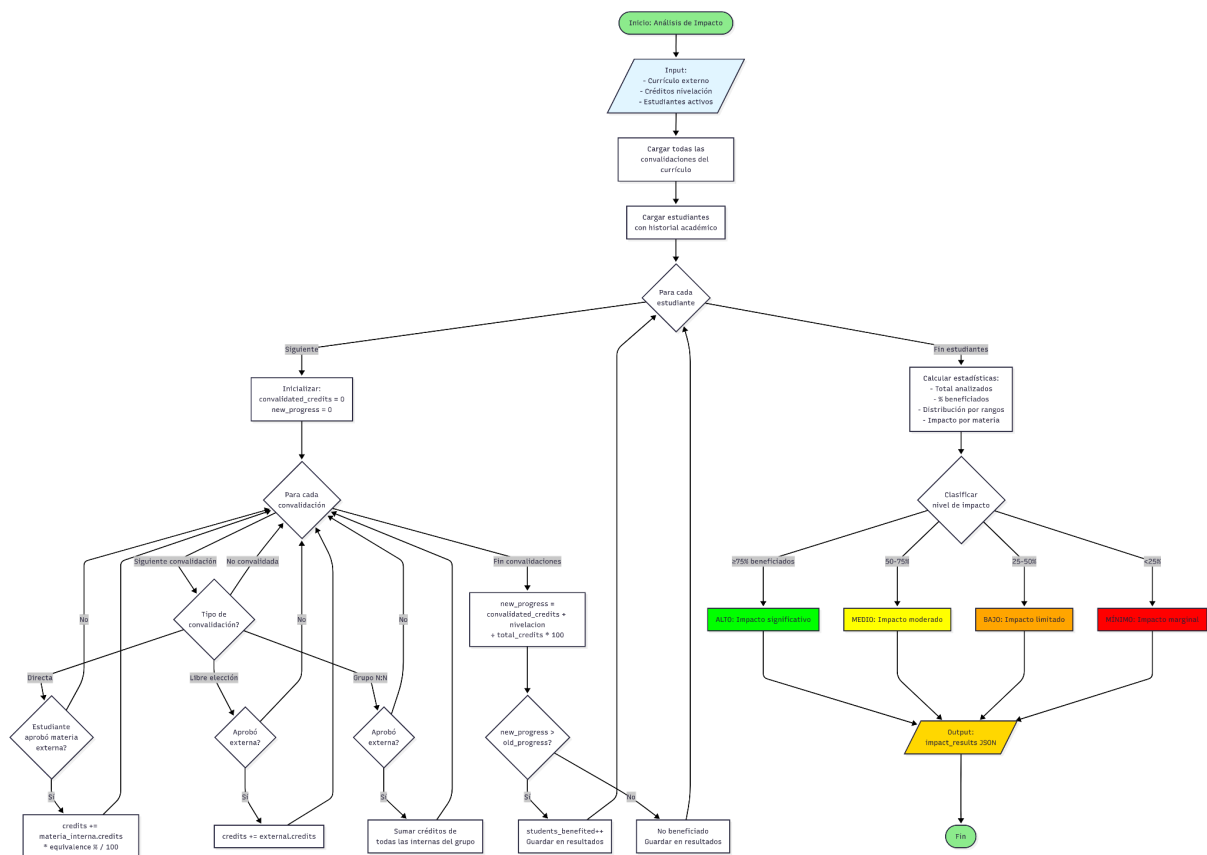
Tipos de Convalidación (Diagrama de Estados)



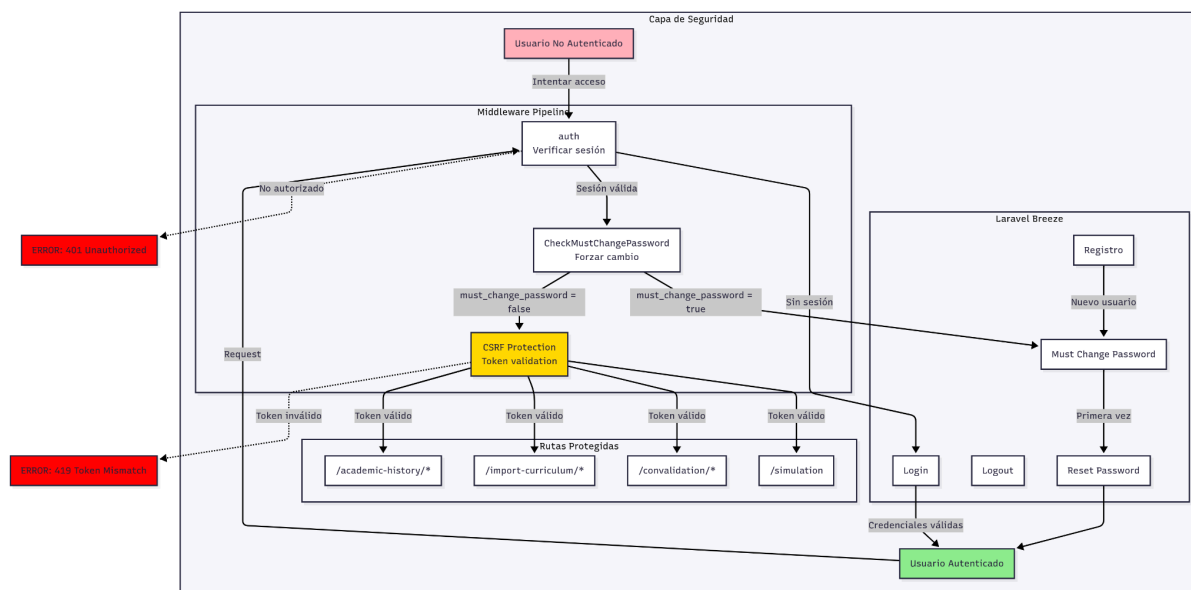
Pipeline de Importación de Currículos (Wizard)



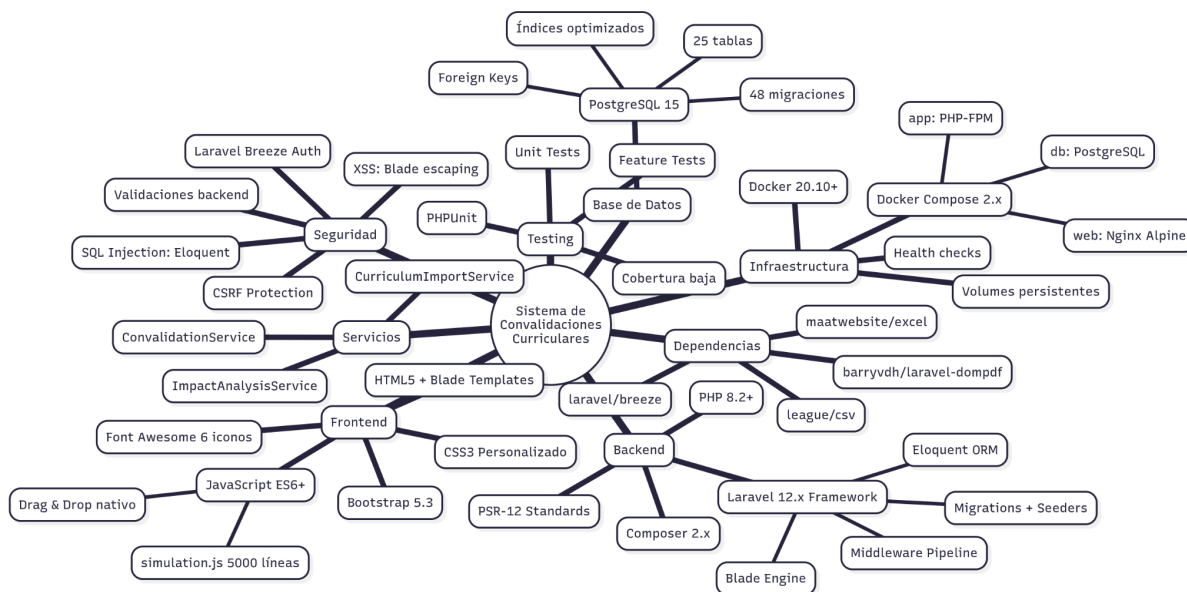
Análisis de Impacto (Algoritmo)



Seguridad y Autenticación



Stack tecnológico completo



Despliegue y Operación

