

UNIVERSIDAD TECNICA DE COTOPAXI

Integrantes: Stiven Cañar, Xavier Moposita, Erick Sacatoro, Kevin Vaca, Bolivar Yugcha

1. TÍTULO DEL SISTEMA

El nombre de nuestro sistema es “Archivum” este es un nombre que hemos elegido para nuestro proyecto de software. Este sistema nos va a permitir gestionar, describir y consultar recursos archivísticos en repositorios institucionales, facilitando la administración de documentos, autoridades y objetos digitales, con control de acceso por roles y soporte para estándares archivísticos.

2. IDEA DEL PROYECTO

Principales Funcionalidades

- Gestión de recursos y colecciones archivísticas
- Administración de usuarios y permisos
- Gestión de objetos digitales
- Sistema de autenticación y autorización
- Búsqueda y consulta de recursos
- Gestión de contenedores y ubicaciones

Características

- Interfaz web responsive
- Sistema multi-usuario con roles
- Base de datos relacional
- Validación de datos en frontend y backend
- API RESTful

Recomendaciones

- Implementar HTTPS en producción
- Realizar backups automáticos de la BD
- Documentar toda la API
- Seguir principios SOLID y Clean Code

3. OBJETIVOS

1. **Objetivo General:** Desarrollar un sistema web para la gestión de archivos digitales que permita a los archivistas organizar, catalogar y administrar colecciones de recursos de manera eficiente.
2. **Objetivos Específicos:**
 - Implementar un sistema de autenticación y autorización con roles
 - Crear módulos para gestionar recursos, accesos y objetos digitales
 - Desarrollar funcionalidades de búsqueda y consulta
 - Establecer un sistema de gestión de contenedores y ubicaciones
 - Garantizar la integridad y seguridad de los datos

4. FECHAS CRÍTICAS

- **Inicio del Proyecto:** Semana 1
- **Diseño de arquitectura:** Semana 1
- **Definición de backlog:** Semana 1
- **Sprint 1:** Semana 2-3
- **Sprint 2:** Semana 4
- **Sprint 3:** Semana 5
- **Sprint 4:** Semana 6

5. MIEMBROS DEL EQUIPO DE DESARROLLO

Distribución de Roles SCRUM

Product Owner: [Erick]

- Define prioridades del backlog
- Valida criterios de aceptación
- Toma decisiones sobre funcionalidades

Scrum Master: [Xavier]

- Facilita ceremonias SCRUM
- Elimina impedimentos
- Asegura seguimiento de prácticas ágiles

Development Team: [Stiven, Kevin, Bolivar]

- Desarrollador Full Stack 1
- Desarrollador Full Stack 2
- Desarrollador Full Stack 3

6. PRODUCT BACKLOG

El Product Backlog del Sistema reúne todas las funcionalidades necesarias para cumplir los objetivos del proyecto. Estas funcionalidades se expresan mediante Historias de Usuario, las cuales están priorizadas según su valor para el negocio y planificadas para ser desarrolladas de forma incremental durante los sprints.

Las historias de mayor prioridad están orientadas a la autenticación, gestión de usuarios y administración de recursos archivísticos, garantizando la base funcional del sistema. Las historias de prioridad media fortalecen la organización, clasificación y localización de los recursos, mientras que las de prioridad baja se enfocan en funcionalidades avanzadas y mejoras de control, como la gestión de roles y la prevención de ediciones concurrentes.

El Product Backlog es gestionado y priorizado por el Product Owner, y se revisa periódicamente durante las sesiones de Backlog Refinement, permitiendo adaptarse a cambios en los requerimientos y asegurar la entrega continua de valor al usuario final.

7. PLANIFICACIÓN DE SPRINTS

SPRINT 1

Objetivo: Establecer fundamentos del sistema - autenticación, usuarios y recursos básicos

Historias de Usuario:

- HU-01: Autenticación de Usuario (5 pts)
- HU-02: Crear Cuenta de Usuario (5 pts)
- HU-03: Crear y Editar Recurso (8 pts)

Total: 18 puntos

Capacidad del equipo: 20 puntos

SPRINT 2

Objetivo: Implementar gestión de objetos digitales y búsqueda

Historias de Usuario:

- HU-04: Gestionar Objetos Digitales (8 pts)

- HU-05: Búsqueda de Recursos y Accesos (5 pts)
- HU-06: Restablecer Contraseña de Usuario (3 pts)

Total: 16 puntos

Capacidad del equipo: 20 puntos

SPRINT 3

Objetivo: Gestión de contenedores, ubicaciones y clasificación

Historias de Usuario:

- HU-07: Asociar Instancia con Contenedor (5 pts)
- HU-08: Asignar Ubicación a Contenedor (5 pts)
- HU-09: Asociar Sujetos con Recursos (5 pts)

Total: 15 puntos

Capacidad del equipo: 20 puntos

SPRINT 4

Objetivo: Completar funcionalidades avanzadas y refinamiento

Historias de Usuario:

- HU-10: Crear y Editar Eventos (5 pts)
- HU-11: Gestión de Roles y Permisos (8 pts)
- HU-12: Prevención de Sobrescritura Concurrente (8 pts)

Total: 21 puntos

Capacidad del equipo: 22 puntos

8. PRÁCTICAS ÁGILES A USAR

1. **Daily Stand-ups:** Reuniones diarias de 15 minutos
2. **Sprint Planning:** Al inicio de cada sprint (2 horas)
3. **Sprint Review:** Al final de cada sprint (1 hora)
4. **Sprint Retrospective:** Al final de cada sprint (1 hora)
5. **Pair Programming:** Para funcionalidades complejas
6. **Code Review:** Revisión de código antes de merge
7. **Continuous Integration:** Integración continua del código

8. **Definition of Done:** Checklist claro para cada historia
9. **Velocity Tracking:** Seguimiento de puntos completados por sprint
10. **Backlog Refinement:** Sesión semanal de refinamiento
11. **Timeboxing:** Todas las reuniones tienen tiempo limitado
12. **Task Board:** Tablero en trello
13. **User Story Mapping:** Mapeo de historias por flujo de usuario

9. TECNOLOGÍAS A USAR

Frontend

- **HTML5:** Estructura de páginas
- **CSS:** Estilos y diseño responsive
- **Bootstrap 5:** Framework CSS
- **JavaScript:** Lógica del cliente

Backend

- **Node.js:** Entorno de ejecución
- **Express.js:** Framework web
- **JWT (jsonwebtoken):** Autenticación
- **bcrypt:** Encriptación de contraseñas
- **dotenv:** Variables de entorno

Base de Datos

- **MySQL:** SGBD relacional
- **MySQL Workbench:** Diseño y administración de BD

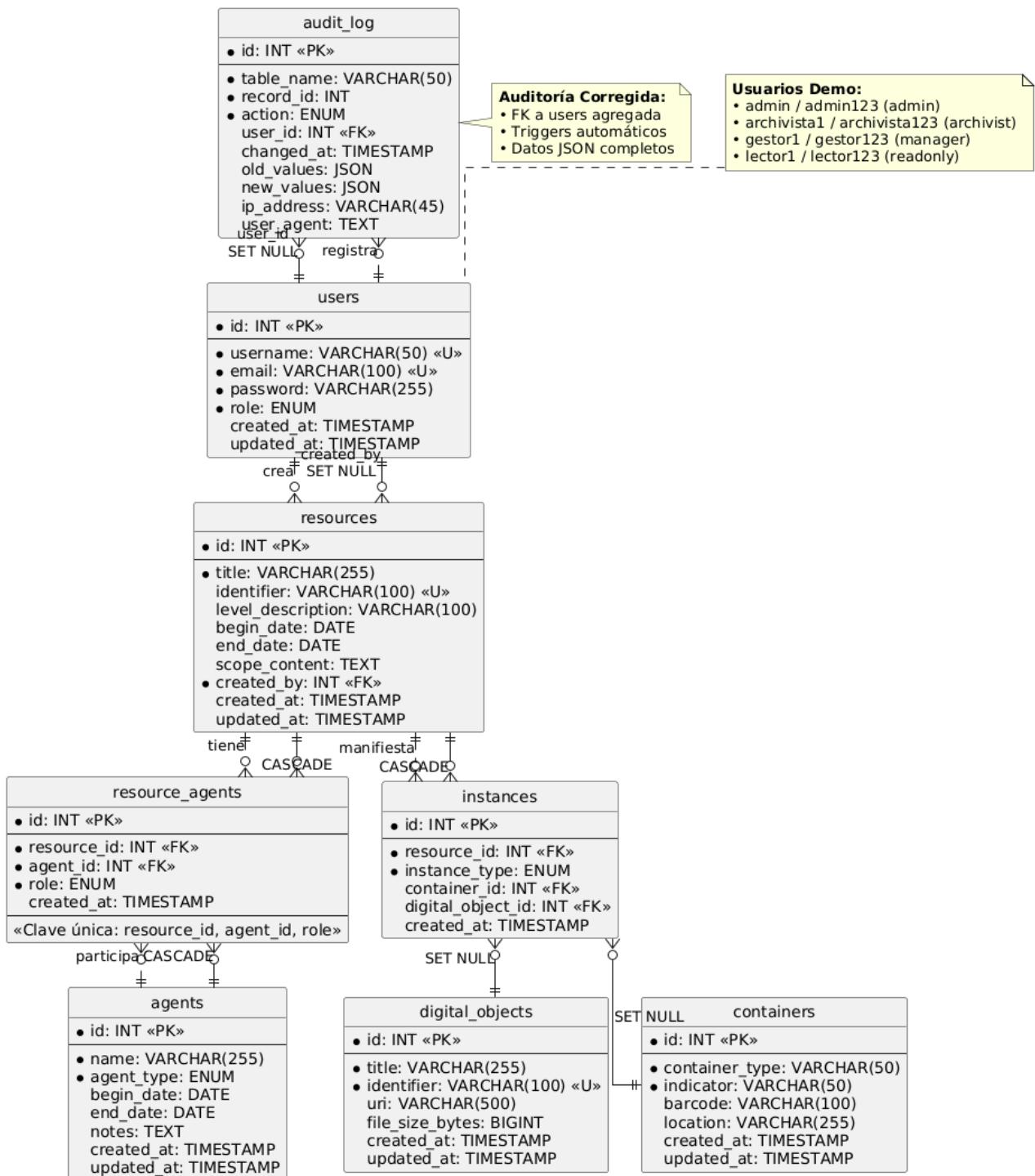
Control de Versiones

- **Git:** Sistema de control de versiones
- **GitHub:** Repositorio remoto

Herramientas de Desarrollo

- **Visual Studio Code:** IDE
- **Postman:** Pruebas de API

10. MODELO DE DATOS (MODELO RELACIONAL)



11. HISTORIAS DE USUARIO

HISTORIAS DE USUARIO SELECCIONADAS Y PRIORIZADAS

PRIORIDAD ALTA

HU-01: Autenticación de Usuario

Como Usuario

Quiero poder iniciar sesión en el sistema

Para acceder a las funcionalidades según mi rol

Puntos de Historia: 5

Sprint: 1

Restricciones:

- La contraseña debe tener mínimo 8 caracteres
- El sistema debe bloquear el usuario después de 5 intentos fallidos
- La sesión debe expirar después de 30 minutos de inactividad

Criterios de Aceptación:

1. El sistema muestra un formulario de login con campos de usuario y contraseña
2. El sistema valida que ambos campos estén completos
3. El sistema verifica las credenciales contra la base de datos
4. Si las credenciales son correctas, redirige al dashboard según el rol
5. Si las credenciales son incorrectas, muestra mensaje de error
6. El botón de login se deshabilita durante la validación

DoD (Definition of Done):

- Código desarrollado y comentado
- Formulario de login responsive
- Validaciones frontend implementadas
- Endpoint de autenticación en Node.js funcionando
- Pruebas de login exitoso y fallido realizadas
- Sesión persistente verificada
- Código subido a repositorio

HU-02: Crear Cuenta de Usuario (Administrador)

Como Administrador

Quiero crear nuevas cuentas de usuario

Para dar acceso al sistema a nuevos miembros

Puntos de Historia: 5

Sprint: 1

Restricciones:

- Solo usuarios con rol de Administrador pueden crear cuentas
- El email debe ser único en el sistema
- El nombre de usuario debe ser único

Criterios de Aceptación:

1. El sistema muestra formulario con campos: nombre, email, usuario, contraseña, rol
2. El sistema valida formato de email
3. El sistema valida que usuario y email no existan previamente
4. El sistema encripta la contraseña antes de guardar
5. El sistema crea el registro en la tabla users
6. El sistema muestra mensaje de confirmación
7. El sistema envía notificación al nuevo usuario (simulada)
8. El nuevo usuario aparece en la lista de usuarios

DoD:

- Formulario de creación desarrollado
- Validaciones implementadas (frontend y backend)
- Contraseña encriptada con bcrypt
- Registro guardado en base de datos
- Lista de usuarios actualizada automáticamente
- Pruebas de validación realizadas

HU-03: Crear y Editar Recurso

Como Archivista

Quiero crear y editar registros de recursos

Para catalogar las colecciones del archivo

Puntos de Historia: 8

Sprint: 1

Restricciones:

- Solo usuarios con rol Archivista o superior pueden crear recursos
- El título es campo obligatorio
- El identificador debe ser único

Criterios de Aceptación:

1. El sistema muestra formulario con campos: título, identificador, descripción, fecha, nivel de descripción
2. El sistema valida campos obligatorios
3. En modo creación, guarda nuevo registro en tabla resources
4. En modo edición, carga datos existentes en el formulario
5. En modo edición, actualiza el registro en base de datos
6. El sistema muestra mensaje de confirmación
7. El sistema redirige a la vista de detalle del recurso
8. El recurso aparece en la lista de recursos del repositorio

DoD:

- Formulario CRUD completo desarrollado
- Validaciones frontend y backend implementadas
- Vista de lista de recursos implementada
- Vista de detalle de recurso implementada
- Pruebas de crear, editar, listar realizadas
- Interfaz responsive verificada

HU-04: Gestionar Objetos Digitales

Como Archivista

Quiero administrar objetos digitales

Para vincular archivos digitales con los recursos

Puntos de Historia: 8

Sprint: 2

Restricciones:

- Los archivos no deben superar 10MB
- Formatos permitidos: PDF, JPG, PNG, DOCX
- Cada objeto digital debe vincularse a un recurso

Criterios de Aceptación:

1. El sistema muestra formulario con campos: título, identificador digital, tipo, archivo, recurso asociado
2. El sistema permite seleccionar archivo desde el equipo
3. El sistema valida tamaño y formato del archivo
4. El sistema guarda el archivo en carpeta /uploads
5. El sistema guarda metadata en tabla digital_objects
6. El sistema vincula el objeto con el recurso seleccionado
7. El sistema muestra vista previa del archivo (si aplica)
8. El sistema permite editar metadata del objeto
9. El sistema permite eliminar objetos digitales

DoD:

- Formulario de carga de archivos implementado
- Validaciones de tamaño y formato funcionando
- Carpeta /uploads creada y configurada
- Vista previa de archivos implementada
- Vinculación con recursos verificada
- Pruebas de carga, edición, eliminaciones realizadas

HU-05: Búsqueda de Recursos y Accesos

Como Archivista

Quiero buscar recursos y accesos dentro del repositorio

Para localizar rápidamente información específica

Puntos de Historia: 5

Sprint: 2

Restricciones:

- La búsqueda debe ser case-insensitive
- Debe buscar en título, identificador y descripción
- Solo muestra resultados del repositorio del usuario

Criterios de Aceptación:

1. El sistema muestra campo de búsqueda en la barra de navegación
2. El sistema busca en tiempo real mientras se escribe (debounce 500ms)
3. El sistema muestra resultados de recursos que coincidan
4. El sistema muestra resultados de accesos que coincidan
5. El sistema resalta el término buscado en los resultados
6. El sistema muestra mensaje "Sin resultados" si no hay coincidencias
7. Al hacer clic en un resultado, redirige a la vista de detalle
8. El sistema limita a 20 resultados por búsqueda

DoD:

- Barra de búsqueda implementada
- Debounce de 500ms configurado
- Endpoint GET /api/search?q=término creado
- Query SQL con LIKE implementada
- Resaltado de términos en resultados
- Componente de resultados de búsqueda creado
- Redirección a detalles funcionando
- Pruebas de búsqueda realizadas

PRIORIDAD MEDIA**HU-06: Restablecer Contraseña de Usuario**

Como Administrador

Quiero restablecer la contraseña de un usuario

Para ayudarlos cuando olviden sus credenciales

Puntos de Historia: 3

Sprint: 2

Restricciones:

- Solo administradores pueden restablecer contraseñas
- La nueva contraseña temporal debe tener 12 caracteres aleatorios
- El usuario debe cambiar la contraseña en el próximo login

Criterios de Aceptación:

1. El sistema muestra botón "Restablecer contraseña" en la lista de usuarios
2. Al hacer clic, muestra modal de confirmación
3. El sistema genera contraseña temporal aleatoria
4. El sistema actualiza la contraseña en la base de datos (encriptada)
5. El sistema marca el usuario con flag "debe cambiar contraseña"
6. El sistema muestra la contraseña temporal al administrador (una sola vez)
7. El sistema muestra mensaje de confirmación
8. En el próximo login, el usuario es obligado a cambiar la contraseña

DoD:

- Botón de restablecimiento agregado
- Modal de confirmación implementado
- Campo must_change_password agregado a tabla users
- Flujo de cambio obligatorio de contraseña implementado
- Pruebas de restablecimiento realizadas
- Seguridad de visualización única verificada

HU-07: Asociar Instancia con Contenedor

Como Archivista

Quiero asociar instancias con contenedores

Para gestionar la ubicación física de los materiales

Puntos de Historia: 5

Sprint: 3

Restricciones:

- Una instancia puede estar en un solo contenedor
- El contenedor debe existir previamente
- El tipo de contenedor debe ser válido

Criterios de Aceptación:

1. El sistema muestra formulario para crear instancia con selector de contenedor
2. El sistema lista contenedores disponibles en dropdown
3. El sistema permite especificar tipo de instancia
4. El sistema valida que el contenedor existe
5. El sistema crea registro en tabla instances vinculando recurso y contenedor
6. El sistema muestra el contenedor asociado en la vista de recurso
7. El sistema permite editar la asociación
8. El sistema permite eliminar la asociación

DoD:

- Formulario de instancias implementado
- Dropdown de contenedores cargado dinámicamente
- Validaciones de existencia implementadas
- Vista de recurso muestra contenedores asociados
- Pruebas de asociación realizadas
- Integridad referencial verificada

HU-08: Asignar Ubicación a Contenedor

Como Archivista

Quiero asignar información de ubicación a contenedores

Para saber dónde están físicamente los materiales

Puntos de Historia: 5

Sprint: 3

Restricciones:

- La ubicación debe tener coordenadas (edificio, sala, estante, nivel)
- Un contenedor puede tener múltiples ubicaciones históricas
- La ubicación actual es la más reciente

Criterios de Aceptación:

1. El sistema muestra formulario con campos: edificio, sala, estante, nivel, coordenadas
2. El sistema valida que todos los campos estén completos
3. El sistema crea registro en tabla locations
4. El sistema vincula la ubicación con el contenedor
5. El sistema marca la ubicación como "actual"
6. El sistema muestra historial de ubicaciones en vista de contenedor
7. Al asignar nueva ubicación, la anterior se marca como "histórica"
8. El sistema muestra mapa visual de la ubicación (opcional)

DoD:

- Formulario de ubicaciones implementado
- Lógica de actualización de ubicación actual implementada
- Vista de historial de ubicaciones creada
- Pruebas de asignación y historial realizadas
- Validaciones de campos obligatorios verificadas

HU-09: Asociar Sujetos con Recursos

Como Archivista

Quiero asociar registros de sujetos con descripciones de recursos

Para clasificar temáticamente los materiales

Puntos de Historia: 5

Sprint: 3

Restricciones:

- Un recurso puede tener múltiples sujetos
- Un sujeto puede asociarse a múltiples recursos
- Los sujetos deben existir previamente en el sistema

Criterios de Aceptación:

1. El sistema muestra campo de selección múltiple de sujetos en formulario de recurso
2. El sistema carga lista de sujetos disponibles
3. El sistema permite seleccionar uno o más sujetos
4. El sistema muestra sujetos asociados en vista de recurso
5. El sistema permite agregar/quitar sujetos en modo edición
6. El sistema permite buscar recursos por sujeto
7. El sistema muestra cantidad de recursos por sujeto

DoD:

- Selector múltiple de sujetos implementado
- Vista de recurso muestra sujetos
- Búsqueda por sujeto implementada
- Contador de recursos por sujeto agregado
- Pruebas de asociaciones N:M realizadas

HU-10: Crear y Editar Eventos

Como Archivista

Quiero crear y editar registros de eventos

Para documentar actividades relacionadas con los recursos

Puntos de Historia: 5

Sprint: 4

Restricciones:

- Todo evento debe tener fecha y tipo
- El evento debe vincularse a al menos un recurso o agente
- Los tipos de evento son predefinidos

Criterios de Aceptación:

1. El sistema muestra formulario con campos: tipo, fecha, descripción, agente, recurso
2. El sistema provee dropdown con tipos de evento predefinidos
3. El sistema valida fecha en formato válido
4. El sistema permite seleccionar recurso y/o agente relacionado
5. El sistema guarda evento en tabla events
6. El sistema muestra eventos en timeline del recurso
7. El sistema permite editar eventos existentes
8. El sistema permite filtrar eventos por tipo y fecha

DoD:

- Formulario de eventos implementado
- Tabla events creada
- Timeline de eventos en vista de recurso implementado
- Filtros de eventos funcionando
- Validación de fechas implementada
- Pruebas de CRUD de eventos realizadas

PRIORIDAD BAJA

HU-11: Gestión de Roles y Permisos

Como Administrador del Sistema

Quiero asignar roles a usuarios

Para controlar el acceso a diferentes funcionalidades

Puntos de Historia: 8

Sprint: 4

Restricciones:

- Los roles son: Admin, Repository Manager, Archivist, Read Only
- Cada rol tiene permisos específicos predefinidos
- Un usuario solo puede tener un rol

Criterios de Aceptación:

1. El sistema muestra lista de usuarios con su rol actual
2. El sistema permite cambiar rol desde dropdown
3. El sistema valida que el rol seleccionado existe
4. El sistema actualiza el rol del usuario en base de datos
5. El sistema aplica permisos según el nuevo rol inmediatamente
6. El sistema muestra matriz de permisos por rol
7. El sistema restringe acceso a funciones según rol
8. El sistema muestra mensaje si el usuario no tiene permisos

DoD:

- Tabla roles creada con roles predefinidos
- Tabla permissions creada
- Selector de rol en gestión de usuarios
- Matriz de permisos documentada
- Pruebas de restricción por rol realizadas

HU-12: Prevención de Sobrescritura Concurrente

Como Archivista

Quiero que el sistema me avise si alguien más está editando un registro

Para no sobrescribir cambios de otros usuarios

Puntos de Historia: 8

Sprint: 4

Restricciones:

- El sistema debe detectar ediciones concurrentes
- El usuario debe poder ver los cambios en conflicto
- Se debe usar control de versiones optimista

Criterios de Aceptación:

1. El sistema agrega campo version a todas las tablas principales
2. Al cargar un registro para edición, guarda el número de versión
3. Al guardar, verifica que la versión no haya cambiado
4. Si la versión cambió, muestra mensaje de advertencia
5. El sistema muestra diálogo con cambios conflictivos
6. El usuario puede elegir: sobrescribir, cancelar o fusionar cambios
7. Al guardar exitosamente, incrementa número de versión
8. El sistema registra en log todos los conflictos detectados

DoD:

- Campo version agregado a tablas principales
- Lógica de versionado optimista implementada
- Diálogo de conflictos implementado
- Opciones de resolución de conflictos funcionando
- Pruebas de edición concurrente realizadas
- Documentación de versionado creada

12. ACRÓNIMOS

Acrónimo	Definición
SGAD	Sistema de Gestión de Archivos Digitales
HU	Historia de Usuario
DoD	Definition of Done (Definición de Terminado)
CRUD	Create, Read, Update, Delete
API	Application Programming Interface
REST	Representational State Transfer
JWT	JSON Web Token
BD	Base de Datos
SGBD	Sistema Gestor de Base de Datos
FK	Foreign Key (Clave Foránea)
PK	Primary Key (Clave Primaria)
N:M	Relación Muchos a Muchos
IDE	Integrated Development Environment
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
JS	JavaScript
HTTPS	HyperText Transfer Protocol Secure
CORS	Cross-Origin Resource Sharing
TDD	Test-Driven Development