

**UNIVERSIDAD TECNOLÓGICA DE EL SALVADOR
FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS
PROGRAMACION IV**



“Documento final: Sistema de Gestión de pasantías universitarias (SIGEPU)”

Docente:

Ing. Rene Antonio Iraheta Reyes

Presentado por:

Aguilar García, Mario Alejandro	25-1667-2023
Deleon Juárez, Katherin Lisseth	25-3833-2023
Gutiérrez De la O, Daniel Mateo	17-0800-2023
Mejía Díaz, Gabriela Michelle	25-1354-2023
Rivera Cortez, Lisbeth Beatriz	25-0631-2023

San Salvador, 7 de diciembre del 2025

Indice

Introducción	1
Justificación.....	2
Objetivos.....	3
Objetivo general	3
Objetivos específicos	3
1. Alcance del proyecto	4
1.1. Alcance Incluido (Lo que si hace el sistema)	4
1.2. Alcance Excluido (Lo que NO hace el sistema - Limitaciones).....	5
2. Tecnologías utilizadas.....	6
3. Base de datos	7
3.1. Diccionario de datos	7
3.2. Diagrama de la base de datos.....	11
4. Conclusiones	12

Introducción

El proceso mediante el cual los estudiantes universitarios buscan oportunidades de pasantías y las empresas reclutan talento joven suele ser complejo, manual y desarticulado. La falta de comunicación directa entre los actores involucrados genera ineficiencia, retrasos y pérdida de oportunidades tanto para estudiantes como para organizaciones.

Con el fin de resolver estas problemáticas, se propone el desarrollo del Sistema de Gestión de Pasantías Universitarias (SIGEPU), una plataforma web que permitirá centralizar, organizar y simplificar el flujo de postulación y aprobación de pasantías. Este documento presenta la definición del Producto Mínimo Viable (MVP), diseñado para validar la funcionalidad esencial del sistema antes de avanzar hacia versiones más completas.

Justificación

El SIGEPU surge como respuesta a la necesidad de contar con un mecanismo más eficiente y transparente para gestionar pasantías universitarias. A través de una interfaz amigable y un flujo de trabajo simplificado, se busca:

1. Reducir la carga administrativa para la universidad.
2. Agilizar la interacción entre estudiantes y empresas.
3. Garantizar una trazabilidad clara de las postulaciones.
4. Validar tempranamente la utilidad del sistema mediante un MVP.
5. El MVP permitirá verificar la aceptación del flujo completo: publicación → postulación →
6. aprobación, sirviendo de base para futuras mejoras y módulos avanzados.

Objetivos

Objetivo general

Desarrollar una aplicación web funcional que implemente el flujo crítico del proceso de pasantías: publicación de ofertas, postulación por parte del estudiante y aprobación universitaria. Esta primera versión permitirá evaluar la efectividad del sistema y orientar su expansión futura.

Objetivos específicos

1. Implementar un sistema seguro de registro y autenticación para los roles: Estudiante, Gestor Empresarial y Gestor Universitario.
2. Permitir la publicación de ofertas de pasantías por parte de las empresas.
3. Proporcionar a los estudiantes una interfaz sencilla para visualizar y postularse a ofertas.
4. Ofrecer al Gestor Universitario un panel básico para aprobar o rechazar postulaciones.
5. Implementar una base de datos en MySQL que soporte el flujo mínimo del sistema.

1. Alcance del proyecto

1.1. Alcance Incluido (Lo que si hace el sistema)

El sistema **SIGEPU** es una aplicación web integral ("Full Stack") funcional que cubre el ciclo completo de la gestión de pasantías:

A. Gestión de Actores y Seguridad

- **Múltiples Roles:** Soporte para tres tipos de usuarios con paneles independientes: **Estudiantes, Empresas y Gestores Universitarios.**
- **Seguridad:** Autenticación robusta con encriptación de contraseñas (BCrypt) y protección de rutas en el Frontend para evitar accesos no autorizados.
- **Perfil de Usuario:** Capacidad para que todos los usuarios suban y actualicen su **foto de perfil** y modifiquen sus datos personales.
- **Integridad de Datos:** Detección automática de usuarios nuevos para obligar el llenado de su perfil académico/empresarial antes de usar el sistema.

B. Módulo de Estudiantes

- **Búsqueda y Filtro:** Visualización de ofertas de pasantía disponibles, con filtrado dinámico por **Carrera.**
- **Postulación Inteligente:** Validación automática de requisitos (carrera afín) y **bloqueo de duplicidad** (no permite aplicar a más de una oferta de la misma empresa simultáneamente).
- **Seguimiento:** Visualización en tiempo real del estado de la solicitud (Pendiente, Aprobado, Contratado, etc.).
- **Gestión de Renuncia:** Posibilidad de solicitar la cancelación de una postulación activa.

C. Módulo de Empresas

- **Gestión de Ofertas:** Publicación, edición y desactivación (ocultamiento) de vacantes de pasantía.
- **Selección de Talento:** Visualización exclusiva de candidatos que ya pasaron el filtro académico de la universidad.
- **Contratación:** Capacidad de marcar a un estudiante como **CONTRATADO** o **DESCARTADO**, cerrando el proceso de selección.

D. Módulo de Gestión Universitaria

- **Validación Académica:** Revisión y aprobación/rechazo de las postulaciones iniciales de los estudiantes.
- **Supervisión Final:** Monitoreo de las pasantías activas (estudiantes ya contratados).

- **Cierre de Proceso:** Facultad para dar por **FINALIZADA** una pasantía y aprobar las horas sociales del estudiante.

E. Automatización de Reglas de Negocio

- **Limpieza Automática:** Cuando un estudiante es contratado por una empresa, el sistema **cancela automáticamente** todas sus otras postulaciones pendientes para liberar esas vacantes.

1.2. Alcance Excluido (Lo que NO hace el sistema - Limitaciones)

Estas son funcionalidades que **no** forman parte de este MVP (Producto Mínimo Viable), aunque podrían agregarse en versiones futuras:

- **Integración con Sistemas Externos:** El sistema no se conecta con la base de datos oficial de la universidad para validar notas o inscripción de materias en tiempo real (funciona con su propia base de datos interna).
- **Notificaciones por Correo Electrónico:** No se envían emails reales a los usuarios (las notificaciones son visuales dentro de la plataforma).
- **Chat Interno:** No existe un sistema de mensajería directa entre Estudiante y Empresa; la comunicación se asume externa (teléfono/email) una vez aprobada la solicitud.
- **Administración de Catálogos:** No hay una interfaz gráfica para crear nuevas Carreras o Usuarios Administradores (estos se gestionan a nivel de Base de Datos/Scripts SQL).
- **Reportes Avanzados:** No se generan gráficos estadísticos o exportaciones a PDF/Excel de los datos.

Este resumen demuestra que tu proyecto es sólido, funcional y tiene límites bien definidos, lo cual es clave para una evaluación académica exitosa.

2. Tecnologías utilizadas

El sistema ha sido construido bajo una arquitectura robusta y escalable, utilizando estándares modernos de la industria:

- **Lenguaje y Backend:** Java 21 + Spring Boot 3 (Implementación de API RESTful).
- **Base de Datos:** MySQL 8 (Relacional, optimizada con tipos de datos BIGINT e integridad referencial).
- **Persistencia de Datos (ORM):** Spring Data JPA y Hibernate (Mapeo Objeto-Relacional automatizado).
- **Seguridad:** Spring Security (Autenticación, Autorización basada en Roles y encriptación BCrypt).
- **Frontend:** HTML5, CSS3 y **Vanilla JavaScript** (Fetch API) diseñados para ofrecer una experiencia de usuario dinámica y fluida (*SPA-like*).
- **Gestión de Dependencias:** Apache Maven.
- **Arquitectura:** Diseño en Capas (Controlador, Servicio, Repositorio y Entidad) para garantizar la modularidad y el mantenimiento del código.

3. Base de datos

3.1. Diccionario de datos

Tabla 1: usuarios

Descripción: Almacena las credenciales de acceso y el rol de seguridad de todos los actores del sistema. Es la tabla principal para la autenticación.

Campo	Tipo de Dato	Restricciones	Descripción
id	BIGINT	PK, AUTO_INCREMENT	Identificador único del usuario.
email	VARCHAR(255)	NOT NULL, UNIQUE	Correo electrónico usado para iniciar sesión. No se puede repetir.
password	VARCHAR(255)	NOT NULL	Contraseña encriptada (Hash BCrypt). Nunca texto plano.
rol	ENUM	NOT NULL	Define los permisos: 'ESTUDIANTE', 'GESTOR_EMPRESARIAL', 'GESTOR_UNIVERSITARIO'.
foto	VARCHAR(255)	NULLABLE	Nombre del archivo de la foto de perfil almacenada en el servidor (ej: uuid_foto.jpg).

Tabla 2: carreras

Descripción: Catálogo de la oferta académica de la universidad. Sirve para filtrar pasantías y validar perfiles.

Campo	Tipo de Dato	Restricciones	Descripción
id	BIGINT	PK, AUTO_INCREMENT	Identificador único de la carrera.
nombre	VARCHAR(255)	NOT NULL	Nombre oficial de la carrera (ej: "Ingeniería en Sistemas").

Tabla 3: estudiantes

Descripción: Contiene la información personal y académica del alumno. Vincula una cuenta de usuario con una carrera específica.

Campo	Tipo de Dato	Restricciones	Descripción
id	BIGINT	PK, AUTO_INCREMENT	Identificador del perfil de estudiante.
nombre	VARCHAR(100)	NOT NULL	Nombre(s) del estudiante.
apellido	VARCHAR(100)	NOT NULL	Apellido(s) del estudiante.
id_usuario	BIGINT	FK, NOT NULL	Relación con la tabla usuarios.
id_carrera	BIGINT	FK, NOT NULL	Relación con la tabla carreras. Define a qué ofertas puede aplicar.

Tabla 4: empresas

Descripción: Almacena la información de las entidades empleadoras.

Campo	Tipo de Dato	Restricciones	Descripción
id	BIGINT	PK, AUTO_INCREMENT	Identificador único de la empresa.
nombre	VARCHAR(255)	NOT NULL	Nombre comercial de la empresa.
id_usuario_gestor	BIGINT	FK, NOT NULL	Relación con la tabla usuarios. Cuenta que administra esta empresa.

Tabla 5: pasantias

Descripción: Representa las ofertas o vacantes publicadas por las empresas.

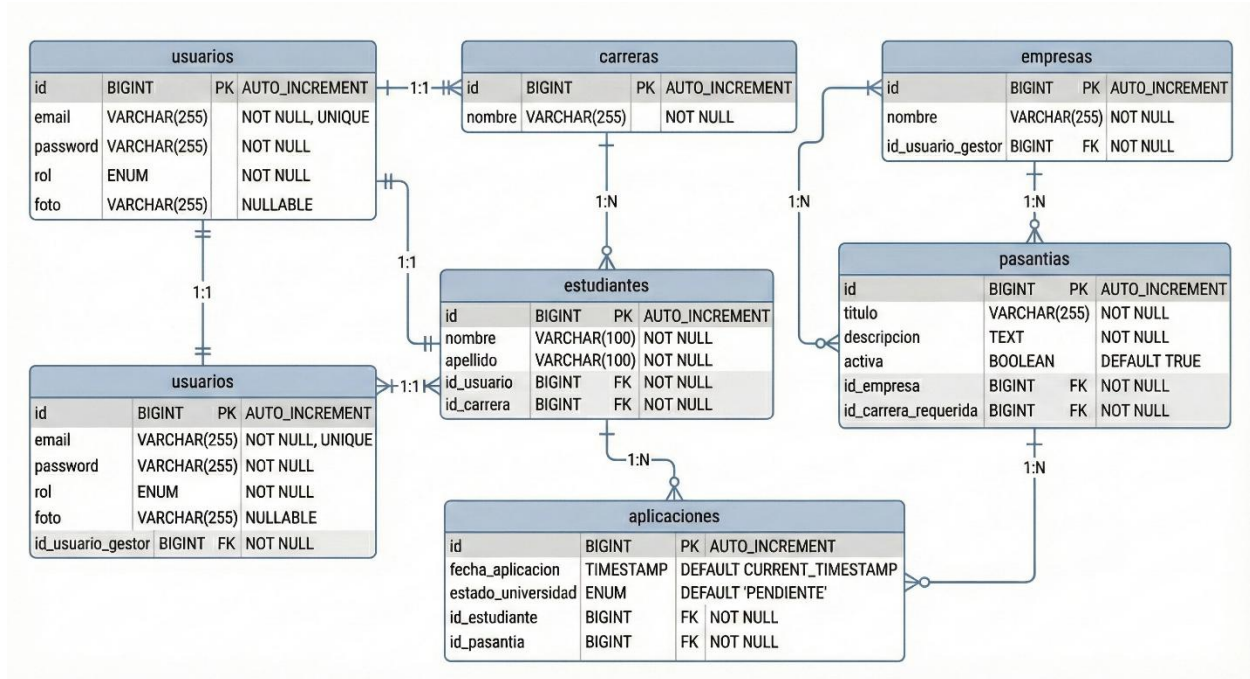
Campo	Tipo de Dato	Restricciones	Descripción
id	BIGINT	PK, AUTO_INCREMENT	Identificador único de la oferta.
titulo	VARCHAR(255)	NOT NULL	Título del puesto (ej: "Desarrollador Java").
descripcion	TEXT	NOT NULL	Detalle de actividades y requisitos.
activa	BOOLEAN	DEFAULT TRUE	Interruptor para mostrar (1) u ocultar (0) la oferta a los estudiantes.
id_empresa	BIGINT	FK, NOT NULL	Empresa que publica la oferta.
id_carrera_requerida	BIGINT	FK, NOT NULL	Carrera requerida para postularse.

Tabla 6: aplicaciones

Descripción: Tabla transaccional que registra la postulación de un estudiante a una pasantía y el estado actual del proceso.

Campo	Tipo de Dato	Restricciones	Descripción
id	BIGINT	PK, AUTO_INCREMENT	Identificador único de la solicitud.
fecha_aplicacion	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Fecha y hora exacta de la postulación.
estado_universidad	ENUM	DEFAULT 'PENDIENTE'	Estado del flujo: 'PENDIENTE', 'APROBADO' (por U), 'RECHAZADO', 'SOLICITUD_CANCELACION', 'CANCELADO', 'CONTRATADO', 'NO_SELECCIONADO', 'FINALIZADO'.
id_estudiante	BIGINT	FK, NOT NULL	Estudiante que aplica.
id_pasantia	BIGINT	FK, NOT NULL	Pasantía a la que se aplica.

3.2. Diagrama de la base de datos



4. Conclusiones

Tras finalizar el desarrollo e implementación del Sistema de Gestión de Pasantías Universitarias (SIGEPU), se han obtenido las siguientes conclusiones:

1. Centralización y Automatización Exitosa

El sistema ha logrado cumplir su objetivo principal al integrar en una única plataforma a los tres actores fundamentales del proceso (Estudiantes, Empresas y Universidad). Se ha eliminado la dependencia de procesos manuales y dispersos, permitiendo un flujo de información continuo y transparente donde cada cambio de estado (postulación, aprobación, contratación) es visible en tiempo real para los involucrados.

2. Robustez de la Arquitectura Tecnológica

La elección de Spring Boot (Java) como backend y MySQL como motor de base de datos ha demostrado ser una decisión acertada para garantizar la escalabilidad y la integridad de los datos. La implementación de una arquitectura en capas (Controlador, Servicio, Repositorio) permite que el sistema sea modular, facilitando el mantenimiento y la futura incorporación de nuevas funcionalidades sin afectar la lógica de negocio existente.

3. Integridad de Datos y Reglas de Negocio

Uno de los mayores logros del proyecto es la implementación de reglas de negocio "inteligentes" que protegen la coherencia del sistema. Funcionalidades como el bloqueo de duplicidad en postulaciones y la cancelación automática de solicitudes pendientes al concretarse una contratación, demuestran que el software no solo almacena datos, sino que gestiona activamente el ciclo de vida de la pasantía, evitando errores humanos y conflictos de interés.

4. Seguridad y Control de Acceso

El sistema cumple con estándares de seguridad modernos mediante la implementación de Spring Security y encriptación BCrypt. Se ha garantizado que la información sensible (como contraseñas) esté protegida y que la navegación esté estrictamente restringida según el rol del usuario, previniendo accesos no autorizados tanto a nivel de Backend (API) como de Frontend (Vistas).

5. Experiencia de Usuario (UX) Funcional

A pesar de no utilizar frameworks de frontend complejos, el uso de HTML5, CSS3 y JavaScript nativo (Fetch API) ha permitido construir una interfaz dinámica y reactiva (SPA-like). La inclusión de perfiles de usuario personalizables (fotos, datos) y la retroalimentación visual inmediata (alertas, cambios de estado en colores) aseguran una curva de aprendizaje baja y una experiencia de uso satisfactoria para el usuario final.