



Informe Final Proyecto Ingeniería de Software
“Prácticas UBB”

Integrantes: Carlos Gómez
Anyelo Gática
Matias Reyes
Nicolás Muñoz
Cristopher Alarcón

Fecha: 29 de Diciembre de 2025

CAPÍTULO I

I.1. DEFINICIONES, SIGLAS Y ABREVIACIONES

- **JWT (JSON Web Token)**: Estándar para la creación de tokens de acceso que permiten la transmisión segura de información entre partes.
- **CRUD**: Acrónimo de Crear, Leer, Actualizar y Borrar (las operaciones básicas de base de datos).
- **Stakeholders**: Actores interesados en el proyecto (Alumnos, Docentes, Jefes de Carrera).
- **Bitácora**: Registro cronológico de actividades que el estudiante debe completar (funcionalidad de compañero).
- **Cupo**: Vacante disponible en una oferta de práctica.

I.2. PRESENTACIÓN DEL CONTEXTO

El presente proyecto se desarrolla en el contexto de la **Educación Superior**, específicamente orientado a la gestión académica de las **Prácticas Profesionales**. Esta etapa es un hito crítico en la formación de los estudiantes, ya que constituye el primer acercamiento formal al mundo laboral y es un requisito obligatorio para la obtención del título profesional.

En el entorno actual de la institución, el proceso de práctica involucra a tres actores fundamentales que deben coordinarse eficientemente:

1. **El Estudiante**: Quien debe buscar, postular y validar su centro de práctica, además de reportar su progreso.
2. **El Docente Encargado**: Responsable de buscar oportunidades en el mercado, publicarlas, validar las postulaciones y supervisar el desempeño académico del alumno.
3. **La Administración Académica**: Que requiere un registro fidedigno de que el estudiante ha cumplido con los requisitos curriculares.

Actualmente, las instituciones educativas se encuentran en un proceso de **transformación digital**, buscando migrar sus procesos administrativos manuales o análogos hacia plataformas web centralizadas. En este escenario, la gestión de la información relativa a las prácticas (ofertas, cupos, bitácoras de avance y retroalimentación) requiere de un sistema que garantice la integridad de los datos, la immediatez en la comunicación y la transparencia en la asignación de cupos, reemplazando los métodos informales como el correo electrónico, tableros físicos o planillas de cálculo desconectadas.

Este software se inserta precisamente en este ecosistema, actuando como el puente tecnológico que digitaliza y estandariza el flujo completo de la práctica profesional, desde la publicación de la oferta hasta el cierre del proceso.

I.3. DESCRIPCIÓN DE LA PROBLEMÁTICA Actualmente, la gestión de prácticas profesionales se realiza de manera dispersa. Los docentes publican ofertas por correo o tablones físicos, lo que provoca que la información se pierda. No existe un control centralizado de quién postula a qué, ni validación automática de cupos. Además, los estudiantes que consiguen prácticas por fuera (externas) no tienen dónde registrarlas formalmente, y la comunicación (feedback) queda aislada en correos personales.

I.4. PROPUESTA DE SOLUCIÓN Desarrollar una plataforma web centralizada que gestione el ciclo de vida completo de la práctica. Desde la publicación de ofertas con control de cupos en tiempo real, pasando por la postulación del estudiante, hasta el seguimiento mediante bitácoras y comunicación directa docente-alumno vía comentarios.

I.6. JUSTIFICACIÓN DEL PROYECTO Este sistema optimiza el tiempo de los docentes al automatizar la selección (listados de postulantes) y notificaciones por correo. Para el estudiante, democratiza el acceso a las oportunidades y centraliza sus antecedentes académicos en un solo lugar.

Capítulo II. DEFINICIÓN DEL PROYECTO

II.1. OBJETIVO GENERAL Implementar un sistema web para la gestión, administración y seguimiento de prácticas profesionales, facilitando la interacción entre estudiantes y docentes encargados.

II.2. OBJETIVOS ESPECÍFICOS

1. Gestionar ofertas de práctica (CRUD) con control automático de cupos.
2. Implementar un módulo de postulación segura que notifique vía correo electrónico a los encargados.
3. Permitir el registro de prácticas externas conseguidas autogestionadamente por los alumnos (Aporte compañero).
4. Habilitar un sistema de bitácoras para el seguimiento del avance del estudiante (Aporte compañero).
5. Establecer un canal de comentarios para feedback directo en la plataforma (Aporte compañero).

II.7. HERRAMIENTAS, FRAMEWORK, LENGUAJE

- **Frontend:** React (Vite), CSS3, SweetAlert2 (para alertas), Axios (conexión API).
- **Backend:** Node.js con Express.
- **Base de Datos:** PostgreSQL (o la que estés usando) gestionada con TypeORM.
- **Seguridad:** Bcrypt (encriptación claves) y JWT (autenticación).
- **Servicios:** Nodemailer (envío de correos SMTP Google).

Capítulo III. DEFINICIÓN DEL SOFTWARE

III.5. REQUERIMIENTOS FUNCIONALES (Resumen)

- **RF-01 Autenticación:** Inicio de sesión diferenciado (Admin, Docente, Estudiante).
- **RF-02 Gestión de Ofertas:** Crear, editar, eliminar y listar ofertas (con paginación).
- **RF-03 Postulación:** El estudiante postula, se descuenta cupo, se valida unicidad y se envía correo.
- **RF-04 Visualización de Postulantes:** El docente puede ver tabla de alumnos inscritos en su oferta.
- **RF-05 Prácticas Externas:** Formulario para ingresar prácticas no listadas en el sistema.
- **RF-06 Bitácora:** Registro de actividades semanales del alumno.
- **RF-07 Feedback:** Sistema de comentarios en las ofertas/prácticas.

III.6. REQUERIMIENTOS NO FUNCIONALES

- **Seguridad:** Las contraseñas no se guardan en texto plano.
- **Usabilidad:** Interfaz responsive y feedback visual inmediato (alertas de carga, éxito y error).
- **Rendimiento:** Carga de listados optimizada mediante paginación.

CAPÍTULO V. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

(Nota: Aquí corregí los títulos que venían en tu plantilla original para que calcen con tu proyecto).

V.1. ESPECIFICACIÓN DE LA SOLUCIÓN Aplicación Web tipo SPA (Single Page Application) basada en arquitectura cliente-servidor RESTful.

V.2. OPTIMIZACIÓN DEL RENDIMIENTO Se implementó **paginación** en el listado de ofertas públicas para evitar sobrecarga de datos en el cliente. Se utiliza **useEffect**

y `useCallback` en React para evitar renderizados innecesarios y consumo excesivo de memoria.

V.3. COMPONENTE: MODELO DE DATOS (En vez de ML/DL) (Si no tienen IA, elimina la sección ML/DL y pon esto): Se diseñó un modelo relacional normalizado.

- Entidad `User`: Roles y datos personales.
- Entidad `OfertaPractica`: Relación "Many-to-Many" con `User` (tabla intermedia de postulaciones) y "Many-to-One" con `User` (encargado).
- (*Menciona aquí las tablas de Bitácora y Comentarios de tus compañeros*).

V.4. SERVIDORES V.4.1. SERVIDOR CON EXPRESS Se utiliza Express.js como framework de backend para manejar las rutas, middlewares de autorización (`isAdmin`, `isEstudiante`) y controladores. (*Elimina la sección V.4.2 de FastAPI si no la usan*).

V.5. SISTEMA WEB DE GESTIÓN DE PRÁCTICAS (Título corregido) V.5.4.

BACKEND Arquitectura en capas:

1. **Rutas**: Definición de endpoints.
2. **Middlewares**: Validación de JWT y Roles.
3. **Controladores**: Lógica de negocio (ej: lógica de descontar cupos).
4. **Servicios**: Comunicación con la BD (Repositories).

V.5.5. FRONTEND Desarrollado en React. Uso de **Hooks** personalizados (`useUsers`, `useOfertas`) para separar la lógica de la vista. Implementación de **Rutas Protegidas** para evitar que estudiantes entren a paneles de administración.

Capítulo VI. EVALUACIONES Y RESULTADOS

Aquí puedes mencionar las pruebas que acabamos de hacer:

- **Prueba de Postulación:** Se verificó que al postular, el cupo disminuye en la base de datos y en la vista sin recargar (React State).
- **Prueba de Unicidad:** El sistema impide postular dos veces a la misma oferta.
- **Prueba de Notificación:** Se comprobó la recepción del correo vía Nodemailer.
- **Prueba de Roles:** Un estudiante no puede ver el botón "Eliminar Oferta", solo "Postular".

Capítulo VII. PLAN DE IMPLANTACIÓN

Este capítulo describe cómo el sistema pasará del entorno de desarrollo (tu computador) al entorno real de la institución, asegurando que los usuarios sepan usarlo.

VII.1. PLAN DE CAPACITACIÓN/ENTRENAMIENTO

Para asegurar la correcta adopción del "Sistema de Gestión de Prácticas", se ha diseñado un plan de capacitación diferenciado según el perfil de usuario:

- **Administradores y Jefes de Carrera:**
 - **Modalidad:** Taller presencial práctico (2 horas).
 - **Contenidos:** Gestión de usuarios (CRUD), restauración de contraseñas, supervisión global de ofertas y generación de reportes.
 - **Material de Apoyo:** Manual técnico en PDF y acceso a un usuario "Super Admin" de prueba.
- **Docentes Encargados:**
 - **Modalidad:** Webinar sincrónico o cápsulas de video (tutoriales).
 - **Contenidos:** Cómo crear una oferta de práctica, revisar la lista de postulantes, descargar datos de alumnos y utilizar el módulo de comentarios para feedback.
 - **Objetivo:** Que el docente pueda publicar una oferta y gestionar sus cupos sin asistencia técnica.
- **Estudiantes:**
 - **Modalidad:** Autoinstrucción digital.
 - **Contenidos:** Infografía paso a paso enviada al correo institucional y un video corto ("Reel" o "Short") en las redes sociales de la carrera explicando cómo postular, subir bitácoras y registrar prácticas externas.
 - **Enfoque:** Uso intuitivo de la interfaz, resaltando la importancia de mantener su perfil actualizado.

VII.1.1. ESTRATEGIA DE IMPLANTACIÓN

La puesta en marcha del sistema se realizará bajo una estrategia de "**Implantación en Fases**" (Phased Rollout) para minimizar riesgos:

Fase 1: Piloto Controlado (Semana 1-2):

- Se desplegará el sistema solo para una carrera específica o un grupo pequeño de docentes "Beta Testers".
- **Objetivo:** Detectar errores no vistos en desarrollo y validar la usabilidad del flujo de postulación.

Fase 2: Ajustes y Feedback (Semana 3):

- Se recopilarán los comentarios del grupo piloto para corregir bugs críticos o mejorar textos confusos en la interfaz.

Fase 3: Masificación (Semana 4 en adelante):

- Apertura del sistema a todas las carreras planificadas.
- Migración de datos antiguos (si aplica) y apagado progresivo de los métodos anteriores (correos masivos, planillas Excel).

Capítulo VIII. CONCLUSIÓN DEL PROYECTO

El desarrollo del **Sistema de Gestión de Prácticas Profesionales** ha permitido dar solución a una problemática administrativa latente en la institución, modernizando un proceso que, hasta ahora, se caracterizaba por la dispersión de la información y la gestión manual.

A través de la implementación de una arquitectura de software robusta basada en el stack **MERN** (en este caso usando PostgreSQL con Node.js y React), se logró cumplir con los objetivos planteados: centralizar las ofertas, automatizar el control de cupos y transparentar el proceso de postulación para los estudiantes.

Técnicamente, el proyecto demostró la viabilidad de integrar funcionalidades complejas como el envío de correos transaccionales (Nodemailer), la seguridad mediante tokens (JWT) y la gestión de relaciones de base de datos (TypeORM) en una interfaz amigable y responsive.

En conclusión, este software no solo optimiza tiempos operativos para los docentes, sino que también entrega a los estudiantes una herramienta formal y digna para gestionar uno de los hitos más importantes de su formación profesional. El sistema queda preparado para escalar y recibir futuros módulos, consolidándose como un activo digital valioso para la comunidad académica.

CAPÍTULO IX. REFERENCIAS

Aquí debes listar la documentación y herramientas que usaste. Te dejo las estándar para este proyecto:

1. **React Documentation.** (2024). *Quick Start*. Recuperado de: <https://react.dev/>
2. **Node.js Foundation.** (2024). *Node.js Documentation*. Recuperado de: <https://nodejs.org/en/docs/>
3. **TypeORM.** (2024). *TypeORM Documentation: Relational Database Object Mapping*. Recuperado de: <https://typeorm.io/>

4. **Express.js.** (2024). *Fast, unopinionated, minimalist web framework for Node.js.* Recuperado de: <https://expressjs.com/>
5. **Nodemailer.** (2024). *Send emails from Node.js.* Recuperado de: <https://nodemailer.com/>
6. **JWT.io.** (2024). *Introduction to JSON Web Tokens.* Recuperado de: <https://jwt.io/introduction>
7. **PostgreSQL.** (2024). *PostgreSQL: The World's Most Advanced Open Source Relational Database.* Recuperado de: <https://www.postgresql.org/docs/>