**Documento de Presentación del Proyecto**

**Proyecto:** Sistema Institucional de Gestión Académica  
**Alumno:** Pedro Emmanuel Márquez  
**Carrera:** Tecnicatura en Desarrollo de Software  
**Materia:** Taller II  
**Año:** 2025

1. **Título del Proyecto**

**Sistema Institucional de Gestión Académica**

1. **Descripción de la problemática**

En los institutos educativos, la gestión de alumnos, docentes, materias e inscripciones suele hacerse de manera manual o en planillas de cálculo dispersas, lo cual provoca errores, pérdida de datos y procesos lentos.

1. **Objetivo de la aplicación web**

* Centralizar la administración de alumnos, docentes, carreras y materias.
* Facilitar el proceso de inscripción de alumnos en materias.
* Contar con un sistema web con backend, base de datos y frontend integrado.

1. **Diagrama entidad-relación y modelo relacional**

**Entidades principales:**

* Alumno (alumnoid, nombre, apellido, carreraid, dni)
* Docente (docenteid, nombre, apellido, titulo, dni)
* Carrera (carreraid, nombre, duracion)
* Materia (materiaid, nombre, horas, carreraid, docenteid)
* Inscripción (inscripcionid, alumnoid, materiaid, fecha)
* AlumnoCarrera (alumnocarreraid, alumnoid, carreraid)
* alumno, docente, carrera, materia, inscripcion, alumnocarrera

**Relaciones:**

* Un alumno pertenece a una carrera.
* Un docente dicta una o varias materias.
* Una materia pertenece a una carrera.
* Un alumno se inscribe en una materia (a través de Inscripción).

1. **Definición de los límites**

Acciones que se podrán realizar en el sistema:

* Administrar alumnos: altas, bajas, modificaciones y consultas.
* Administrar docentes.
* Administrar carreras.
* Administrar materias.
* Administrar inscripciones (registro de qué alumno cursa cada materia).

Límites: no se incluye login de usuarios ni mensajería automática en esta primera versión.

1. **Estimación de tiempos de ejecución (aproximado)**

* Optimista: 3 semanas.
* Pesimista: 6 semanas.
* Promedio considerado: 4 a 5 semanas.

1. **Diseño de la base de datos**

Se utilizará PostgreSQL con modelo relacional.  
Se crearán tablas para Alumno, Docente, Carrera, Materia e Inscripción con claves primarias y foráneas para mantener integridad.

1. **Backend**

Se desarrollará con Node.js y Express, exponiendo una API REST que permita CRUD (alta, baja, modificación y consulta) sobre las entidades.

1. **Pruebas con Postman**

Cada endpoint del backend será probado con Postman para verificar las operaciones GET, POST, PUT y DELETE.

1. **Frontend**

Se realizará con HTML, CSS y JavaScript, permitiendo al usuario final interactuar con el sistema a través de botones y formularios simples.

1. **Documentación**

Se elaborará un documento con:

* Explicación del diseño.
* Diagrama de base de datos.
* Listado de endpoints.
* Guía breve de uso.

1. **Capacitación**

Se capacitará a los usuarios mediante una breve presentación práctica, explicando cómo cargar alumnos, materias, docentes e inscripciones, y cómo consultar los datos en el sistema.