

# Proyecto Final

Mediante los conocimientos adquiridos en las diferentes materias de la carrera se propone la realización de un proyecto que integre las buenas prácticas de programación, análisis y diseño de una base de datos, despliegue de una API que sirva de backend de nuestro sistema Front y cualquier otra documentación que respalde el funcionamiento y comportamiento de las diferentes partes que integran el proyecto.

Para lograr una mejor supervisión por parte de los tutores en cada uno de los proyectos, se establecen las siguientes tecnologías para el empleo de lenguajes de programación, herramientas, motores de bases de datos y cualquier otro servicio que intervenga en el proceso de desarrollo.

- Lenguaje de Programación:
  - Front: REACT (opcional Typescript)
  - Backend API:
    - **Node Js**
    - Express + sequelize
- Base de datos: Mysql, PostgreSQL, MongoDB
- Plataformas de servicios de despliegue en la nube. Render, Netlify, Vercel.
- Repositorios: Github

El trabajo final deberá ser realizado en equipo, en grupos de 2 a 5 estudiantes como máximo.

Cada grupo definirá un responsable que será el encargado de comunicarse con el docente para las entregas iniciales y finales del proyecto. Este responsable puede ser cambiado o sustituido previa notificación al docente.

## Cronograma:

- Generación de un informe descriptivo con la propuesta de trabajo a realizar **(30/09)**
- Aprobación del informe **(5/10)**
- Entrega de prototipos **(30/11)**
  - Frontend (Generación de un prototipo navegable, no funcional)
  - Backend (Utilización de mocks locales o mockapi.io)
  - Generación de documentación de los repositorios (README.md)
  - Generación de tickets o tareas por integrante en un tablero (Ej: Jira)

## Objetivos:

- Integrar conocimientos y competencias en el desarrollo de un producto o un proyecto de software, simulando una experiencia laboral real.
- Aplicar técnicas, metodologías y herramientas de desarrollo para generar un producto o proyecto según las especificaciones de un cliente real o de su inventiva propia.
- Integrar equipos de trabajo colaborativos donde participe de la elaboración, planificación y diseño técnico de un producto o proyecto de desarrollo de Software aplicando las metodologías de sistemas vigentes.
- Aplicar herramientas de gestión colaborativa y ágil de proyectos de uso cotidiano en el mercado laboral actual.

## Criterios de evaluación:

- Manejo de repositorios para ambos prototipos.
- Configuración de servicios de deploy públicos del prototipo.
- Simulación de entorno local de ambos repositorios
- Colección en postman para pruebas contra la API
- Manejo de tableros en Jira para la división de tareas. Cronograma/Sprint
- Documentación del proyecto + Documentación de Repositorios (REDME.md)
- Estandarización del código (linter)
- Testing de componentes. (Solo para la Aprobación)
- Interfaces/Modelos de los datos de nuestra BD tanto en la API como en la respuesta de las peticiones en el front. (Solo para la Aprobación)

## Ampliación de los criterios

- Cursada
  - Repositorio del frontend y backend (Prototipos)
  - Documentación.
    - Colección en postman de nuestra API
    - Manejo de archivos README.md para la instalación de sus repositorios
  - Diagrama Entidad-Relación (ER): Antes de comenzar con la implementación de la base de datos, deberán diseñar un Diagrama Entidad-Relación que represente de manera clara las entidades, sus atributos y las relaciones entre ellas. Este diagrama debe entregarse como parte de la documentación del proyecto.
  - Despliegue de nuestra back y front: Realizar el despliegue de nuestro back y/o front en plataformas que permitan acceder a nuestros proyectos mediante una URL pública. Ej : Vercel, Render, Netlify, Heroku, etc.

- Documentación técnica:
  - Introducción y objetivos del proyecto
  - Descripción de la arquitectura y diseño (diagramas, si corresponden)
  - Tecnologías, librerías y frameworks utilizados
  - Manual de instalación y ejecución
  - Consideraciones finales o limitaciones
- Aprobación:
  - Backend: Deberán desarrollar una API que maneje las operaciones necesarias para la obtención, inserción, actualización y eliminación de datos  
  
La API deberá incorporar protección en TODAS sus rutas mediante una API\_KEY para que ningún usuario externo pueda consultar, insertar o borrar registros de nuestra BD.
  - Conexión con una base de datos: La API deberá estar conectada a una base de datos donde se almacenará la información. Deberán elegir una base de datos relacional o no relacional (por ejemplo, MySQL, PostgreSQL, MongoDB, etc.) y asegurarse de implementar las operaciones CRUD (Create, Read, Update, Delete) de manera eficiente.  
  
En caso de no contar con un servicio gratuito en la nube para base de datos se puede utilizar sqlite3 . Por ej, puede ser incluida en la API de NodeJS y mantener sus datos persistentes en un archivo en disco.
  - Frontend conectado al backend: Finalmente, deben desarrollar un frontend que consuma la API que hayan creado. Este frontend debe mostrar en pantalla los datos devueltos por la API y permitir la interacción necesaria para la gestión de los mismos (por ejemplo, visualizar, agregar o eliminar información).
  - Generación de reportes: El proyecto deberá permitir generar reportes en PDF/XLS no solo de las tablas la base de datos sino con de información relevante para el usuario.  
  
Ej: Si tenemos 3 tablas de personas, eventos y reservas, debemos poder emitir un reporte detallado de reservas por persona, por evento o global.

## Entregables Generales:

- Repositorio del backend (API). (Repo en GitHub)
- Repositorio del frontend que consuma la API. (Repo en GitHub)
- Diagrama Entidad-Relación de la base de datos.
- Base de datos en formato SQL o el tipo asociado a el motor elegido
- Documentación técnica que explique brevemente el funcionamiento de cada uno de los componentes.

## **Entregables globales del sistema:**

- Login / Logout / Recupero de contraseña.
- Registro (si corresponde)
- Home + estadísticas principales del sistema.
- Header, Menú/Sidebar y Footer.
- Perfil del usuario. (Cambio de datos personales, actualización de Email y contraseña)
- Tabla + ABM de Usuarios (\*)
- Estructuración de directorios del proyecto + Estandarización de carpetas / componentes

## **Entregables del sistema por Alumno:**

- Listado de Registros + Búsqueda + Paginado + Filtro con cantidad de elementos
- Visualización de un registro individual de una lista.
- Operaciones de actualización o eliminación sobre la lista o registro individual.
- Consultas por pantalla + reporte
- Generación de reportes para listados completo o filtrado.
- (\*) Puede tomarse esta sección como un entregable por alumno

Este proyecto les permitirá afianzar los conceptos fundamentales de la arquitectura de software, el manejo de APIs, la integración entre frontend y backend y por sobre todo la experiencia de trabajar con otros programadores reflejando así un entorno similar al utilizado en empresas de software. Recuerden que estaré disponible para cualquier duda o consulta que puedan tener durante el desarrollo del trabajo.

¡Éxitos a todos!

Saludos,  
Sebastian Gañan