



## **TRABAJO FINAL DE MÁSTER: DESARROLLA TU PROYECTO PERSONAL**

Este no es solo un ejercicio técnico: es el primer paso hacia tu futuro profesional.

El mercado digital demanda ideas frescas, productos que resuelvan problemas reales y soluciones que marquen la diferencia. Tu misión no es solo aprobar, sino crear algo con verdadero potencial: una aplicación Full Stack innovadora, funcional y escalable, que pueda convertirse en tu carta de presentación... o incluso, en el germen de tu propio proyecto empresarial.

### **POR QUÉ ESTE PROYECTO ES DIFERENTE**

- No es un ejercicio académico: es un trampolín profesional.
- Puede ser el primer paso para un producto real (¿quién sabe si tu idea atraerá inversores?).
- Demostrarás no solo que sabes programar, sino que entiendes el mercado.

### **ELIGE TU BATALLA: DESARROLLA ALGO QUE TE APASIONE**

No queremos solo código. Queremos tu visión.

- ¿Te apasiona el fitness? Crea una app de entrenamiento y seguimiento avanzado.
- ¿Te mueve la sostenibilidad? Diseña una plataforma para reducir el desperdicio alimentario.
- ¿Eres un gamer? Desarrolla un hub social para jugadores con chat en tiempo real.

La temática es tuya. La tecnología, también. El límite lo pones tú.

### **RECUERDA:**

- Elige una temática que te motive profundamente (pasarás horas con ella).
- Piensa en cómo tu solución podría tener demanda real.
- Prioriza un MVP sólido, pero deja puertas abiertas a futuro. (Producto mínimo viable)

Este es tu momento. No desarrolles solo para aprobar: desarrolla para impactar.

## OBJETIVO

Desarrollar un producto digital original con una idea innovadora que pueda tener incluso impacto en el mercado laboral, implementando una aplicación web Full Stack.

Desarrollaremos un FRONT que defina una interfaz clara, limpia e intuitiva, con una experiencia de usuario funcional y agradable interacción del cliente.

Esto será respaldado con un BACKEND que definirá una API que servirá en la gestión de envío y peticiones para la persistencia de datos que serán almacenados en una base de datos no relacional.

El proyecto se desplegará en producción para que pueda ser accesible por usuarios y clientes.

## REQUISITOS TÉCNICOS:

*Se aceptan librerías de terceros de forma complementaria.*

### Frontend (React)

- Scaffolding bien estructurado:
  - Uso de carpetas como pages, components, hooks, layouts, services, etc.
- Gestión de estado:
  - Correcto uso del estado local y global (mediante Redux o Context API).
  - Implementación de hooks avanzados (useRef, useReducer, useMemo, etc.).
- Autenticación JWT:
  - Sistema de login/registro con tokens JWT y rutas protegidas.  
**(Opcional)**
- Routing:
  - Definición de rutas con React Router
- Consumo de APIs y consulta de datos
  - Gestión de peticiones con axios
  - Manejo de estados con React Query **(Opcional)**

- Testing:
  - Implementación de pruebas unitarias y/o de integración (con Jest, React Testing Library, etc.). **(Opcional)**
- Responsive Design:
  - Diseño adaptable a móvil, tablet y desktop (Mobile First).
- Despliegue:
  - Publicación en un servicio como Vercel, Netlify, GitHub Pages o similar.

## **Backend (Node.js + Express + MongoDB)**

- Scaffolding modular y SOLID:
  - Estructura clara con controllers, services, routes, middlewares, models, etc.
- Base de datos (MongoDB con Mongoose):
  - Modelado de datos y relaciones.
  - Operaciones CRUD (crear, leer, actualizar y eliminar datos).
- Autenticación y seguridad:
  - JWT para autenticación. **(Opcional)**
  - Encriptación de contraseñas (con bcrypt).
  - Validación de datos y protección contra ataques (CORS, rate limiting, sanitización).
- APIs RESTful:
  - Endpoints bien definidos con manejo de errores.
- Funcionalidades adicionales:
  - Subida y gestión de archivos (Multer + FS). **(Opcional)**
  - Envío de emails (Nodemailer). **(Opcional)**
  - Operativa en tiempo real con WebSockets (ej: notificaciones, chat). **(Opcional)**
- Despliegue:
  - Publicación en Render, Railway, AWS, Heroku o similar.

## **FUNCIONALIDAD MÍNIMA REQUERIDA**

- La aplicación debe permitir visualizar, crear, editar y eliminar datos desde la interfaz gráfica, interactuando con la base de datos a través de la API.
- Debe resolver un problema real o aportar valor innovador en alguna temática (ej: salud, educación, ocio, productividad, sostenibilidad, etc.).

## **CRITERIOS DE EVALUACIÓN**

- **Funcionalidad (40%)** - Cumplimiento de los requisitos y operación sin errores.
- **Calidad del código (25%)** - Código bien estructurado, mantenible y siguiendo buenas prácticas.
- **Diseño y experiencia de usuario (15%)** - Interfaz intuitiva y adaptable.
- **Despliegue y versionado (10%)** - Uso adecuado de Git/GitHub y despliegue en servidores.
- **Presentación (10%)**- Explicación clara del proyecto y las tecnologías utilizadas.