# EJERCICIOS DE EVALUACIÓN TÉCNICA

## Desarrollador Full Stack Senior .NET / Vue.js con IA

## EJERCICIO 1: CONVERSIÓN DE PROTOTIPO HTML A VUE.JS CON INTEGRACIÓN DE APIs

Se le proporcionará al candidato:

1. **Prototipo HTML estático**: Un archivo HTML con CSS que representa la interfaz de usuario de una página
2. **Documentación de APIs REST**: Especificaciones de los endpoints, métodos, parámetros y respuestas esperadas
3. **URLs de las APIs**: Endpoints funcionales para realizar las integraciones

### Tarea

Convertir el prototipo HTML en una **aplicación Vue.js 3 funcional** conectada a los webapis proporcionados.

Puedes utilizar **cualquier herramienta de vibe coding** de tu preferencia: RooCode, Claude Code, Cline, Kilo Code, GitHub Copilot, Cursor, Aider, OpenCode, etc.

### Tiempo Estimado

| **Enfoque** | **Tiempo Estimado** |
| --- | --- |
| **Desarrollo Tradicional** (escribiendo código manualmente) | 4-6 horas |
| **Vibe Coding** (usando herramientas de IA efectivamente) | 20-30 minutos |

### Entregables

Video demáximo 5 minutos que muestre como está hecho el Código, que herramienta de vibecoding se utilizó, que prompts se utilizaron y muestre la página funcionando.

## 

## EJERCICIO 2: APLICACIÓN DE ANÁLISIS DE REQUERIMIENTOS CON IA GENERATIVA

**⚡ IMPORTANTE**: Este ejercicio debe hacerse con **vibe coding** en 1 hora máximo. La estructura del código es **totalmente libre**. No se busca arquitectura compleja, solo que funcione.

### Descripción del Ejercicio

Crear una **página web simple** que tenga un cuadro de texto donde el usuario capture la especificación de un software que require y que la IA analice la necesidad y proponga los procesos, subprocesos y casos de uso para cumplir con dicha necesidad, estos procesos, subprocesos y casos de uso deben ser insertados en una base de datos.

### Requisitos Técnológicos

Puedes utilizer cualquier stack tecnológico para la página y la base de datos donde se guardará la información.

**Api IA**

Deberás utilizar la api de openrouter ( <https://openrouter.ai> ) y utilizar cualquier modelo gratuito:

[DeepSeek V3.1 (free)](https://openrouter.ai/deepseek/deepseek-chat-v3.1:free), [OpenAI: gpt-oss-20b (free)](https://openrouter.ai/openai/gpt-oss-20b:free), [Z.AI: GLM 4.5 Air (free)](https://openrouter.ai/z-ai/glm-4.5-air:free) , [Qwen: Qwen3 Coder 480B A35B (free)](https://openrouter.ai/qwen/qwen3-coder:free). etc.

**Base de datos:**

-- Tabla: proceso  
CREATE TABLE proceso (  
 id\_proceso INT IDENTITY(1,1) PRIMARY KEY,  
 nombre NVARCHAR(150) NOT NULL,  
 descripcion NVARCHAR(MAX)  
);  
  
-- Tabla: subproceso  
CREATE TABLE subproceso (  
 id\_subproceso INT IDENTITY(1,1) PRIMARY KEY,  
 id\_proceso INT NOT NULL,  
 nombre NVARCHAR(150) NOT NULL,  
 descripcion NVARCHAR(MAX),  
 FOREIGN KEY (id\_proceso) REFERENCES proceso(id\_proceso)  
);  
  
-- Tabla: caso\_uso  
CREATE TABLE caso\_uso (  
 id\_caso\_uso INT IDENTITY(1,1) PRIMARY KEY,  
 id\_subproceso INT NOT NULL,  
 nombre NVARCHAR(150) NOT NULL,  
 descripcion NVARCHAR(MAX),  
 actor\_principal NVARCHAR(150),  
 tipo\_caso\_uso SMALLINT CHECK (tipo\_caso\_uso IN (1, 2, 3)),  
 -- 1=Funcional, 2=No Funcional, 3=Sistema  
 precondiciones NVARCHAR(MAX),  
 postcondiciones NVARCHAR(MAX),  
 criterios\_de\_aceptacion NVARCHAR(MAX),  
 FOREIGN KEY (id\_subproceso) REFERENCES subproceso(id\_subproceso)  
);

### Tiempo Estimado

| Enfoque | Tiempo Estimado |
| --- | --- |
| **Desarrollo Tradicional** (escribiendo código manualmente) | 8-12 horas |
| **Vibe Coding** (usando herramientas de IA efectivamente) | **45-60 minutos** |

**Nota**: Con vibe coding efectivo y estructura de código libre (sin arquitectura compleja), este ejercicio debería completarse en **1 hora máximo**. El enfoque es lograr la funcionalidad, no crear arquitectura enterprise.

### Entregables

Video demáximo 5 minutos que muestre como está hecho el Código y muestre la página funcionando.