



Power|Mas

**CASO DE EVALUACION  
PRACTICANTE PROGRAMADOR FULL STACK**

**Enero de 2026**



## Tabla de Contenido

<b>1. DESCRIPCIÓN.....</b>	<b>3</b>
<b>2. MODELO DE DATOS: .....</b>	<b>3</b>
<b>3. MODELO DE DATOS: .....</b>	<b>3</b>
<b>4. VALIDACIÓN CONDICIONAL DE DOCUMENTO:.....</b>	<b>4</b>
<b>5. LISTADO DE DOCUMENTOS DE IDENTIDAD:.....</b>	<b>4</b>
<b>6. CONTROLADORES Y ACCIONES:.....</b>	<b>4</b>
<b>7. ENTREGABLES: .....</b>	<b>5</b>



## Caso de Evaluación

### 1. Descripción

Desarrollar un Sistema de Gestión de Beneficiarios que permita registrar, consultar, editar y eliminar beneficiarios de un programa social multi-país..

### 2. Objetivo:

Demostrar habilidades en:

- Desarrollo frontend con React + Typescript
- Desarrollo backend con .NET Web API
- Manejo de base de datos SQL Server
- Validaciones condicionales
- Código limpio y organizado:

### 3. Modelo de Datos:

Define un modelo de datos para los productos que debe incluir al menos los siguientes campos:

Tabla: DocumentoIdentidad

Campo	Tipo	Descripción
Id	int	Primary Key, auto-increment
Nombre	varchar(50)	Nombre del documento (DNI, Pasaporte, etc.)
Abreviatura	varchar(10)	Código corto (DNI, PAS, CE, etc.)
País	varchar(50)	País donde aplica
Longitud	int	Cantidad de caracteres requeridos
SoloNumeros	bit	Si acepta solo números
Activo	bit	Estado del registro



Tabla: Beneficiario

Campo	Tipo	Descripción
Id	int	Primary Key, auto-increment
Nombres	varchar(100)	Nombres del beneficiario
Apellidos	varchar(100)	Apellidos del beneficiario
DocumentoidentidadId	int	FK → Documentoidentidad
NumeroDocumento	varchar(20)	Número de documento
FechaNacimiento	date	Fecha de nacimiento
Sexo	char(1)	M = Masculino, F = Femenino

#### 4. Validación Condicional de Documento:

Cuando el usuario seleccione un tipo de documento, el campo NumeroDocumento debe:

- Validar longitud: Segundo la columna Longitud del tipo de documento
- Validar formato: Si SoloNumeros = true, solo aceptar dígitos
- Mostrar feedback: Indicar al usuario qué formato se espera

#### 5. Listado de Documentos de Identidad:

Crea una página web con las siguientes secciones:

- Obtener lista de documentos desde el backend
- Mostrar en un dropdown/select en el formulario
- Filtrar solo los activos

#### 6. Controladores y Acciones:

Frontend (React)

- Crear proyecto con Vite
- Estructura de carpetas organizada
- Componentes reutilizables
- Manejo de estado (useState, useEffect, o librería de preferencia)
- Formularios con validación
- Estilos CSS (usar TailwindCSS)
- Llamadas HTTP a la API (fetch o axios)



## Backend (.NET)

- Crear proyecto Web API con .NET
- Endpoints RESTful para CRUD de Beneficiarios
- Endpoint para listar Documentos de Identidad
- Conexión a SQL Server
- Manejo de errores básico

## Base de Datos (SQL Server)

- Script para crear las tablas
- Script para insertar datos de ejemplo en DocumentoIdentidad
- Usar Stored Procedures

**7. Entregables:**

1. Código fuente en repositorio Git (GitHub)
2. README.md con:
  - a. Instrucciones de instalación
  - b. Instrucciones para ejecutar el proyecto
  - c. Scripts de base de datos o instrucciones
3. Screenshots o video corto del sistema funcionando