

## **EVALUACIÓN TÉCNICA POSTULANTES ÁREA DE DESARROLLO**



## ÍNDICE

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Proyecto 1 – Desarrollo con Zend Framework.....</b>	<b>3</b>
2.1 Instalación requerida.....	3
2.2 Desarrollo de módulo personalizado.....	3
2.3. Consideraciones técnicas.....	4
2.4 CSS personalizado (Nuevo requerimiento).....	4
<b>4. Recomendaciones adicionales (opcionales).....</b>	<b>4</b>



## 1. Introducción

El presente documento detalla las instrucciones, lineamientos técnicos y entregables requeridos para la evaluación de los postulantes al área de desarrollo.

El objetivo de esta prueba es medir competencias prácticas en instalación de entornos, configuración de frameworks, desarrollo de módulos personalizados, manejo de bases de datos, validaciones, uso de patrones de diseño, buenas prácticas y organización del código.

## 2. Proyecto 1 – Desarrollo con Zend Framework

### 2.1 Instalación requerida

El postulante deberá instalar:

- PHP 8.3.15
- PostgreSQL 17.4.2
- Zend Framework
- Entorno local (XAMPP, WAMP, Docker o similar)

### 2.2 Desarrollo de módulo personalizado

Se debe crear un módulo en Zend Framework con un CRUD que incluya:

- Texto simple
- Alfanumérico
- Correo electrónico
- Fecha
- Hora
- Fecha y hora
- Lista desplegable (mínimo 5 opciones)
- Teléfono
- Carga y visualización de imagen
- Decimal (monto, porcentaje)

**Validaciones:**

- Frontend: Validaciones obligatorias en JavaScript.
- Backend: Validaciones replicadas en servidor.

**2.3. Consideraciones técnicas**

- Uso obligatorio de POO.
- Implementación del patrón MVC.
- Código ordenado, legible y con separación de responsabilidades.
- Base de datos con diseño adecuado y tipos correctos.

**2.4 CSS personalizado (Nuevo requerimiento)**

Se debe incluir un archivo CSS propio, diferente al que viene con Zend. Debe contener:

- Cambios visibles en colores, tipografías o espaciados.
- Estilos propios para formularios, botones y tablas.
- Debe integrarse en el layout principal.

**4. Recomendaciones adicionales (opcionales)**

- Relaciones entre módulos/tablas.
- Mejoras de interfaz.
- Componentes dinámicos simples con JavaScript.
- Uso de ORM o prepared statements.
- Documento de arquitectura (1 página).