

# Prueba de Diagnóstico

---

## Descripción:

Se deberá desarrollar un **Sistema de Registro de productos**. La interfaz deberá validar y guardar los datos del formulario en una base de datos.

Las tecnologías a utilizar serán:

- **HTML**  
La interfaz se deberá desarrollar con HTML utilizando estructuras estándar.
- **CSS**  
La interfaz se deberá estilizar con CSS nativo, no deberá utilizarse frameworks de CSS(Eje: **Bootstrap y Tailwind**)
- **PHP**  
El desarrollo del aplicativo debe estar programado con PHP limpio, sin framework u otro.
- **JavaScript / Ajax**  
Para la conexión entre el HTML y PHP se deberán generar funciones en Javascript(**Javascript nativo**)
- **BASE DE DATOS**  
Los datos deberán ser almacenados(Cual gestor de Base de Datos), de preferencia **PostgreSQL** pero no es obligatorio.

La interfaz deberá tener los campos como se ve en la imagen:

# Formulario de Producto

Código

PROD01K

Nombre

Set Comedor

Bodega

Bodega 1

Sucursal

Sucursal 2

Moneda

DÓLAR

Precio

1500

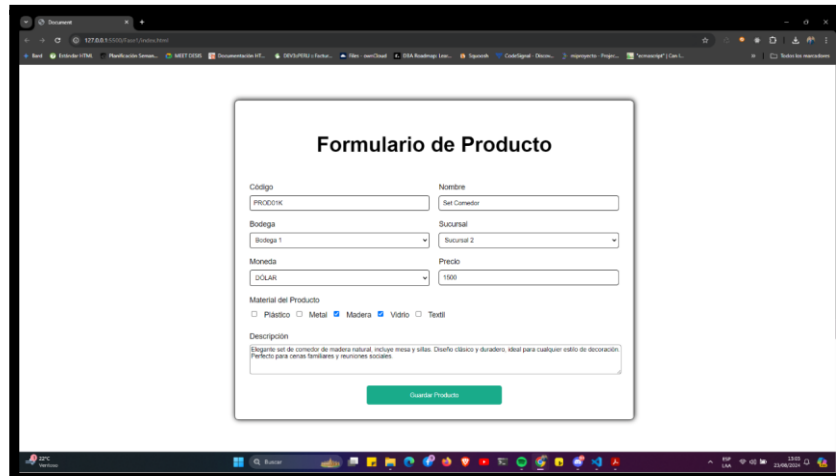
Material del Producto

☐ Plástico ☐ Metal ☒ Madera ☒ Vidrio ☐ Textil

Descripción

Elegante set de comedor de madera natural, incluye mesa y sillas. Diseño clásico y duradero, ideal para cualquier estilo de decoración. Perfecto para cenas familiares y reuniones sociales.

Guardar Producto



## Estructura del Formulario:

El formulario de registro de productos debe contener los siguientes campos:

1. **Código del Producto (input tipo texto)**

Para dudas fmellafarias@gmail.com

零 Validaciones: Obligatorio, formato específico (letras, números), longitud mínima de 5 y máxima de 15 caracteres.

- Debe ser único y verificado en la base de datos.

## 2. Nombre del Producto (input tipo texto)

零 Validaciones: Obligatorio, longitud mínima de 2 y máxima de 50 caracteres.

## 3. Bodega (select)

零 Este campo debe ser cargado dinámicamente desde la base de datos y deberá tener una opción en blanco por defecto.

## 4. Sucursal (select)

零 Dependiendo de la bodega seleccionada, este campo debe cargarse dinámicamente. La primera opción debe ser una opción en blanco.

## 5. Moneda (select)

零 Este campo debe cargarse automáticamente desde la base de datos, con una opción en blanco como primera opción.

## 6. Precio (input tipo texto)

零 Validaciones: Obligatorio, formato de número positivo con hasta dos decimales.

## 7. Material del Producto (grupo de checkboxes)

零 Se debe seleccionar al menos dos opciones de materiales.

## 8. Descripción del Producto (textarea)

零 Validaciones: Obligatorio, longitud mínima de 10 caracteres y máxima de 1000 caracteres.

---

## Estilos CSS:

Para mantener una interfaz simple y funcional, se aplicarán los siguientes estilos:

### Fuente General:

- Estilo de letra: Arial, sans-serif;
- Tamaño de letra: 16px;

### Botón "Guardar Producto":

Color #1aab8a;

Tamaño de letra: 16px;

/\* Ajustar el tamaño y estilo según la imagen de referencia \*/

Este estilo se aplicará al botón de guardado de producto para que sea consistente con el diseño requerido.

---

## Validaciones del Formulario de Registro de Producto:

### 1. Código del Producto:

- Obligatorio: Este campo no debe quedar en blanco.
- Validación JavaScript: La validación de campo obligatorio debe realizarse mediante JavaScript y no con el atributo required en HTML.
- Formato: Debe contener al menos una letra y un número, y no debe contener otros caracteres diferentes.
- Longitud: Debe tener entre 5 y 15 caracteres.
- Unicidad: El código del producto debe ser único, es decir, no debe repetirse al registrar un nuevo producto.
- Errores Personalizados:
  - Si el campo está vacío: `ALERT("El código del producto no puede estar en blanco.")`
  - Si no cumple con el formato (estar compuesta con letras y números): `ALERT("El código del producto debe contener letras y números")`
  - Si no cumple con la longitud: `ALERT("El código del producto debe tener entre 5 y 15 caracteres.")`
  - Si ya existe en la base de datos: `ALERT("El código del producto ya está registrado.")`
- Uso de Regex: La validación de formato debe realizarse mediante expresiones regulares (regex) para asegurar que cumpla con las reglas establecidas.

### 2. Nombre del Producto:

- Obligatorio: Este campo no debe quedar en blanco. Validación por JavaScript.
- Descripción: Ingresar el nombre completo del producto, asegurándose de que sea claro y descriptivo.
- Longitud: Debe tener entre 2 y 50 caracteres.
- Errores Personalizados:
  - Si el campo está vacío: `ALERT("El nombre del producto no puede estar en blanco.")`
  - Si no cumple con la longitud: `ALERT("El nombre del producto debe tener entre 2 y 50 caracteres.")`

### **3. Precio del Producto:**

- Obligatorio: Este campo no debe quedar en blanco. Validación por JavaScript.
- Formato: El precio debe ser un número positivo, con hasta dos decimales (por ejemplo, 19.99).
- Errores Personalizados:
  - Si el campo está vacío: `ALERT("El precio del producto no puede estar en blanco.")`
  - Si no cumple con el formato: `ALERT("El precio del producto debe ser un número positivo con hasta dos decimales.")`
- Uso de Regex: La validación del formato del precio debe realizarse mediante regex para asegurar que siga las reglas especificadas.

### **4. Material del Producto:**

- Obligatorio: Se debe seleccionar al menos dos opciones de materiales disponibles mediante casillas de verificación (checkboxes).
- Errores Personalizados:
  - Si no se seleccionan al menos dos materiales: `ALERT("Debe seleccionar al menos dos materiales para el producto.")`

### **5. Bodega:**

- Obligatorio: Este campo debe seleccionarse. Validación por JavaScript.
- Cargado Automáticamente: Las opciones de la bodega se cargarán automáticamente desde la base de datos.
- Primera Opción Vacía: La primera opción debe ser una opción en blanco.

- Errores Personalizados:
  - Si el campo está vacío: ALERT("Debe seleccionar una bodega.")

#### **6. Sucursal:**

- Condicional: Este campo se cargará automáticamente en función de la bodega seleccionada.
- Cargado Dinámico: Las sucursales disponibles se mostrarán solo si se ha seleccionado una bodega válida.
- Primera Opción Vacía: La primera opción en la lista de sucursales debe ser una opción en blanco.
- Errores Personalizados:
  - Si no se selecciona una sucursal: ALERT("Debe seleccionar una sucursal para la bodega seleccionada.")

#### **7. Moneda:**

- Obligatorio: Este campo debe seleccionarse. Validación por JavaScript.
- Cargado Automáticamente: Las opciones de la moneda se cargarán automáticamente desde la base de datos.
- Primera Opción Vacía: La primera opción debe ser una opción en blanco.
- Errores Personalizados:
  - Si el campo está vacío: ALERT("Debe seleccionar una moneda para el producto.")

#### **8. Descripción del Producto:**

- Obligatorio: Este campo no debe quedar en blanco. Validación por JavaScript.
- Longitud: La descripción debe tener al menos 10 caracteres y no más de 1000, permitiendo cualquier carácter (incluidos espacios y saltos de línea).
- Errores Personalizados:
  - Si el campo está vacío: ALERT("La descripción del producto no puede estar en blanco.")
  - Si no cumple con la longitud: ALERT("La descripción del producto debe tener entre 10 y 1000 caracteres.")

## Flujo de Guardar Producto:

El flujo para guardar un producto sigue los siguientes pasos:

1. **Validación del Formulario:**
  - Cada campo del formulario será validado mediante JavaScript antes de enviar los datos al servidor.
  - En caso de que la validación falle, se mostrarán los mensajes de error personalizados.
2. **Envío mediante AJAX:**
  - Una vez que todos los campos han sido validados correctamente, los datos del formulario serán enviados al servidor mediante AJAX para su procesamiento.
3. **Procesamiento en PHP:**
  - PHP recibirá los datos y luego procederá a insertar los datos en la tabla correspondiente si todas las validaciones son exitosas.
4. **Respuesta al Usuario:**
  - Después de procesar la solicitud, se devolverá una respuesta al cliente (éxito o error) para notificar al usuario sobre el resultado de la operación.

Este flujo asegura que la aplicación funcione de manera fluida y eficiente, proporcionando una experiencia de usuario coherente y amigable.

---

## Archivos a Entregar:

1. **Archivo HTML o PHP:** Contiene el maquetado de la aplicación web y la estructura de la interfaz de usuario.
  2. **Archivo(s) CSS:** Define los estilos y la apariencia visual de la aplicación web, incluyendo colores, tipografías y disposición de elementos.
  3. **Archivo(s) JavaScript:** Maneja la lógica de la aplicación, captura de datos mediante AJAX, y cualquier interacción dinámica en la interfaz de usuario.
  4. **Archivo(s) PHP:** Realiza la comunicación con la base de datos y maneja las operaciones del servidor.
  5. **Archivo(s) SQL:** Contiene las sentencias para la creación de tablas, funciones y otros objetos en la base de datos.
-

## Se evaluará:

Tener en cuenta la “**Rúbrica de evaluación**” adjunta a la evaluación.

---

## Tiempo Estimado:

3 días

## Entrega:

- El código del desarrollo deberá ser entregado mediante una URL pública de GitHub o similar.
- Deberá contener una carpeta SQL con los archivos sql de la base de datos.
- En la raíz deberá existir un archivo LEEME.txt el cual indique los pasos para poder instalar el proyecto
- Adicional deberá indicar la versión de PHP y Base de datos.
- No se aceptarán proyectos comprimidos.
- Para realizar la entrega del proyecto se deberá enviar correo electrónico a [fmellafarias@gmail.com](mailto:fmellafarias@gmail.com) indicando:
  - Nombre Completo
  - RUT / DNI(Perú)
  - Edad
  - URL de GitHub o similar
  - Teléfono

*¡Nos pondremos en contacto contigo en cuanto  
revisemos tu código!*