

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

Ingeniería en Sistemas de Información	
<b>Cátedra:</b> Paradigmas y lenguajes de programación III	<b>Profesor:</b> Mgter. Ing. Agustín Encina
<b>Alumno:</b> Marcos Agustin Varela Ropat	<b>Fecha:</b> 29/10/2025

Duración máxima: 2.30 horas 🖥️

**Instrucciones Generales:**

- Este examen es interactivo y se compone de varias decisiones que tomarás a lo largo del camino.
- Siga las instrucciones cuidadosamente en cada punto de decisión.
- La puntuación total se basará en las decisiones tomadas y en la implementación de las tareas relacionadas con cada opción.
- **No se permiten consultas en línea ni colaboración con otros estudiantes ni con un transformador generativo preentrenado.**

📖 **NARRATIVA DE LA AVENTURA**

Has sido contratado por una startup tecnológica que necesita urgentemente un proyecto web funcional. Tu misión es demostrar tus habilidades como desarrollador full-stack navegando por diferentes desafíos. Cada decisión que tomes definirá tu camino y las tecnologías que dominarás.

**¡Tu reputación como desarrollador está en juego!** 🎯

🛠️ **PARTE 1: DESAFÍOS TEÓRICOS (20 puntos)**

🎲 **ELECCIÓN DE MISIÓN INICIAL**

Antes de comenzar tu proyecto, el equipo técnico necesita evaluar tus conocimientos fundamentales. Elige tu ruta de especialización:

**Elige tu Proyecto (*tildar la opción que vas a desarrollar*):**

- Ruta A: desarrolla el grupo A de preguntas.
- Ruta B: desarrolla el grupo B de preguntas.

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

🌐 **RUTA A: "El Arquitecto Web" (Fundamentos y Estructura)**

**Desafío 1 - Arquitectura de la Web (5 puntos)**

*El CTO te pregunta durante la reunión inicial...*

Dibuja y explica detalladamente la arquitectura Cliente-Servidor. Incluye:

- Componentes principales
- Flujo de comunicación
- Protocolos involucrados
- Ejemplo práctico con un caso real

**Desafío 2 - Maestría en CSS (5 puntos)**

*El diseñador UX necesita claridad en la nomenclatura...*

Explica la diferencia entre selectores de clase y selectores de ID en CSS:

- ¿Cuándo usar cada uno?
- Nivel de especificidad
- Proporciona 2 ejemplos prácticos de cada uno aplicados a una interfaz real

**Desafío 3 - Fundamentos de JavaScript (5 puntos)**

*El líder técnico evalúa tu comprensión de JS...*

Explica el concepto de variables en JavaScript:

- Propósito y utilidad
- Diferencias entre *var*, *let* y *const*
- Proporciona 3 ejemplos mostrando scope y hoisting

**Desafío 4 - Introducción a PHP (5 puntos)**

*El backend developer senior te hace una pregunta clave...*

¿Qué es PHP y cuál es su rol en el desarrollo web moderno?

- Características principales
- Diferencias con lenguajes frontend
- Ejemplo de código PHP integrado en HTML (procesamiento de formulario)

## Examen Interactivo de PLP III: La Aventura del Desarrollador Web 🎮

### 🚀 RUTA B: "El Innovador Técnico" (Evolución y Arquitectura)

#### Desafío 1 - Evolución del HTML (5 puntos)

*El product manager pregunta sobre tecnologías modernas...*

Explica las diferencias clave entre HTML y HTML5:

- Nuevas etiquetas semánticas
- Mejoras en accesibilidad y SEO
- ¿Cómo HTML5 revolucionó el desarrollo web?

**Rta:** Claves: HTML5 agrega semántica (<header> <nav> <main> <article> <section> <aside> <footer>), APIs nativas (audio/video sin plugins, <canvas>, localStorage/sessionStorage, postMessage, history, geolocalización), nuevos inputs (email, date, number, range, color, url, tel), validación nativa, atributos ARIA mejor integrados.

Accesibilidad y SEO: etiquetas semánticas facilitan navegación por lectores y jerarquía clara.

Revolución: elimina dependencias de plugins, estandariza multimedia y gráficos, y habilita apps web ricas.

#### Desafío 2 - Arquitectura CSS Avanzada (5 puntos)

*El tech lead quiere saber si conoces buenas prácticas...*

Explica la diferencia entre arquitectura y metodología en CSS:

- Menciona al menos UNA arquitectura (ej: ITCSS, SMACSS, Atomic)
- Menciona al menos UNA metodología (ej: BEM, OOCSS, SUIT)
- ¿Por qué son importantes en proyectos grandes?

**Rta:** Arquitectura: como se organiza el CSS del proyecto, Ej: ITCSS (Layers: Settings → Tools → Generic → Elements → Objects → Components → Utilities).

Metodología: convenciones para clases. Ej.: BEM (bloque\_\_elemento--modificador) para componibilidad y bajo acoplamiento.

Porque importan: escalabilidad, menor deuda CSS, estilos previsibles en equipos grandes, menos conflictos de especificidad.

## Examen Interactivo de PLP III: La Aventura del Desarrollador Web 🎮

### Desafío 3 - JavaScript vs PHP (5 puntos)

*El arquitecto de software evalúa tu visión técnica...*

Compara y contrasta JavaScript y PHP:

- Diferencias fundamentales (ejecución, tipado, uso)
- 3 escenarios donde JavaScript es más apropiado
- 3 escenarios donde PHP es más apropiado
- Ejemplo de código de cada uno

**Rta:** Diferencias:

- JS: se ejecuta en navegador (y servidor con Node), single-thread + event loop, tipado dinámico; ideal para interactividad UI.
- PHP: del lado servidor, request/response, tipado dinámico, pero con tipos estrictos opcionales, ideal para render server-side y CRUD.

3 escenarios para JS: SPAs reactivas, validación y UX inmediata, tiempo real (WebSockets).

3 escenarios para PHP: render SSR, endpoints REST con SQL, paneles CRUD con sesiones.

Ejemplos:

JS: Sumar items del carrito

```
const total = items.reduce((acc, it) => acc + it.qty * it.price, 0);
```

PHP: Endpoint simple JSON

```
header('Content-Type: application/json');
```

```
echo json_encode(['ok' => true, 'time' => time()]);
```

### Desafío 4 - Conexión a Bases de Datos (5 puntos)

*El DBA necesita confirmar tus conocimientos de persistencia...*

Describe los conceptos fundamentales para conectar PHP con una Base de Datos:

- Métodos de conexión (MySQLi vs PDO)
- Pasos para establecer conexión
- Manejo de errores
- Ejemplo de código con consulta preparada

### **Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

Rta: MySQLi vs PDO: MySQLi solo MySQL, PDO soporta múltiples motores y prepared statements consistentes.

Pasos: crear DSN/ conexión, manejar errores, prepared statements, ejecutar y mapear resultados, cerrar.

Ejemplo PDO + preparada:

```
try {  
  
    $pdo = new PDO('mysql:host=127.0.0.1;dbname=tt_plp3;charset=utf8mb4','root','',[  
        PDO::ATTR_ERRMODE=>PDO::ERRMODE_EXCEPTION  
    ]);  
  
    $stmt = $pdo->prepare('SELECT * FROM tt_products WHERE category = ? LIMIT 20');  
  
    $stmt->execute([$ _GET['cat'] ?? 'all']);  
  
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);  
  
} catch (Throwable $e) {  
  
    http_response_code(500);  
  
    exit('DB error');  
  
}
```

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

🎯 **PARTE 2: PROYECTO PRÁCTICO (80 puntos - *distribuidos en 4 niveles*)**

📖 **Nivel 1 : ELECCIÓN DE PROYECTO BASE (20 puntos)**

**⚠ REGLA CRÍTICA DE NOMENCLATURA:** Todos los archivos, carpetas, clases, funciones, tablas, etc., deben usar como prefijo tus iniciales.

*Ejemplo: Si eres María González López (MGL):*

- 📁 Carpeta: `mgl_assets/`
- 📄 CSS: `mgl_estilos.css`
- 🗄 Base de datos: `mgl_parcial_plp3`
- 📊 Tabla: `mgl_usuarios`
- ⚙ Función: `function mgl_validar()`
- 🖼 Imagen: `mgl_logo.png`
- 🎯 Clase CSS: `.mgl-header`

🎯 **MISIÓN PRINCIPAL - Elige tu Proyecto (*tildar la opción que vas a desarrollar*):**

- Opción A: "MusicStream" - Plataforma de Música Online
- **Opción B: "FoodExpress" - Sistema de Pedidos Online.**
- Opción C: "QuizMaster" - Plataforma de Trivia.

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

🎵 **PROYECTO A: "MusicStream" - Plataforma de Música Online**

*Una discográfica indie quiere su propia plataforma de streaming*

**Requisitos Funcionales:**

- Catálogo de álbumes/canciones con reproductor básico (mínimo 8 items)
- Sistema de búsqueda por artista, género o álbum
- Formulario de suscripción con validación
- Listas de reproducción o favoritos (almacenadas en BD)
- Panel para agregar/editar canciones (CRUD)

**Requisitos No Funcionales:**

- Mínimo 3 secciones distintas (header, galería, formulario)
- Diseño responsive (3 breakpoints)
- Navegación intuitiva y accesible
- Código comentado y estructura modular

## Examen Interactivo de PLP III: La Aventura del Desarrollador Web 🎮

### 🎮 PROYECTO B: "FoodExpress" - Sistema de Pedidos Online

*Un restaurante local necesita digitalizar sus pedidos*

#### Requisitos Funcionales:

- Menú de productos con categorías (mínimo 10 productos)
- Carrito de compras dinámico con subtotales
- Formulario de pedido que guarda en BD
- Sistema de filtrado por categoría
- Panel administrativo para gestionar productos

#### Requisitos No Funcionales:

- Mínimo 3 secciones (menú, carrito, checkout)
- Responsive design con mobile-first
- Feedback visual en todas las interacciones
- Tiempo de carga optimizado

AR\_Parcial\_PLP3/ (Utilizaré AR/ar para las iniciales de mi nombre)

```
├─ index.php
├─ admin.php
├─ css/AR_estilos.css
├─ js/AR_script.js
├─ includes/AR_conexion.php
├─ database/
|   └─ AR_estructura.sql
|   └─ AR_datos.sql
├─ api/
|   └─ products_list.php
|   └─ order_create.php
|   └─ admin_product_save.php
└─ README.md
```



**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

**database/AR\_estructura.sql**

```
CREATE DATABASE IF NOT EXISTS ar_plp3 CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
USE ar_plp3;

CREATE TABLE ar_products (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(120) NOT NULL,
  category VARCHAR(50) NOT NULL,
  price DECIMAL(10,2) NOT NULL,
  stock INT NOT NULL DEFAULT 100,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE ar_orders (
  id INT AUTO_INCREMENT PRIMARY KEY,
  customer_name VARCHAR(120) NOT NULL,
  customer_phone VARCHAR(40) NOT NULL,
  customer_address VARCHAR(200) NOT NULL,
  subtotal DECIMAL(10,2) NOT NULL,
  total DECIMAL(10,2) NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE ar_order_items (
  id INT AUTO_INCREMENT PRIMARY KEY,
  order_id INT NOT NULL,
  product_id INT NOT NULL,
  qty INT NOT NULL,
  unit_price DECIMAL(10,2) NOT NULL,
  FOREIGN KEY (order_id) REFERENCES AR_orders(id),
  FOREIGN KEY (product_id) REFERENCES AR_products(id)
);
```

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

**database/AR\_datos.sql**

```
USE ar_plp3;
INSERT INTO ar_products (name, category, price, stock) VALUES
('Muzzarella', 'Pizzas', 6500, 50),
('Napolitana', 'Pizzas', 7200, 40),
('Calabresa', 'Pizzas', 7200, 35),
('Hamburguesa Clásica', 'Burgers', 5200, 60),
('Hamburguesa Doble', 'Burgers', 6200, 50),
('Papas Fritas', 'Sides', 2500, 100),
('Gaseosa 500ml', 'Bebidas', 1800, 200),
('Agua 500ml', 'Bebidas', 1500, 200),
('Empanadas x3', 'Empanadas', 3000, 100),
('Tarta de Jamón y Queso', 'Tartas', 4800, 25);
```

**includes/AR\_conexion.php**

```
<?php
$AR_DB = null;
try {
    $AR_DB = new PDO('mysql:host=127.0.0.1;dbname=ar_plp3;charset=utf8mb4','root',[
        PDO::ATTR_ERRMODE=>PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE=>PDO::FETCH_ASSOC
    ]);
} catch (Throwable $e) {
    http_response_code(500);
    exit('Error de conexión');
}
```

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

**api/products\_list.php**

```
<?php
require __DIR__.'../includes/AR_conexion.php';
$cat = $_GET['category'] ?? 'all';
if ($cat === 'all') {
    $stmt = $AR_DB->query('SELECT id,name,category,price,stock FROM ar_products
ORDER BY category,name');
} else {
    $stmt = $AR_DB->prepare('SELECT id,name,category,price,stock FROM ar_products
WHERE category=? ORDER BY name');
    $stmt->execute([$cat]);
}
echo json_encode($stmt->fetchAll());
```

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

**api/order\_create.php**

```
<?php
require __DIR__.'../includes/AR_conexion.php';
$input = json_decode(file_get_contents('php://input'), true);
if (!$input || empty($input['items']) || empty($input['customer'])) { http_response_code(400);
exit('Bad request'); }

$items = $input['items']; // [{id, qty, price}]
$customer = $input['customer']; // {name, phone, address}

$AR_DB->beginTransaction();
try {
    // Calcula totales en servidor
    $subtotal = 0.0;
    foreach ($items as $it) {
        $subtotal += (float)$it['price'] * (int)$it['qty'];
    }
    $total = $subtotal; // Aquí podrías sumar delivery o descuentos

    $stmt = $AR_DB->prepare('INSERT INTO tt_orders
(customer_name,customer_phone,customer_address,subtotal,total) VALUES (?, ?, ?, ?, ?)');
    $stmt->execute([$customer['name'], $customer['phone'], $customer['address'], $subtotal,
$total]);
    $orderId = (int)$AR_DB->lastInsertId();

    $stmtItem = $AR_DB->prepare('INSERT INTO tt_order_items
(order_id,product_id,qty,unit_price) VALUES (?, ?, ?, ?)');
    foreach ($items as $it) {
        $stmtItem->execute([$orderId, (int)$it['id'], (int)$it['qty'], (float)$it['price']]);
    }

    $AR_DB->commit();
    echo json_encode(['ok'=>true,'order_id'=>$orderId,'total'=>$total]);
} catch (Throwable $e) {
    $AR_DB->rollBack();
    http_response_code(500);
    echo json_encode(['ok'=>false]);
}
```

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

**api/admin\_product\_save.php**

```
<?php
require __DIR__.'../includes/AR_conexion.php';
$input = json_decode(file_get_contents('php://input'), true);
if (!$input) { http_response_code(400); exit; }
$id = $input['id'] ?? null;
if ($id) {
    $stmt = $AR_DB->prepare('UPDATE ar_products SET
name=?,category=?,price=?,stock=? WHERE id=?');
    $stmt->execute([$input['name'],$input['category'],$input['price'],$input['stock'],$id]);
} else {
    $stmt = $AR_DB->prepare('INSERT INTO ar_products (name,category,price,stock)
VALUES (?, ?, ?, ?)');
    $stmt->execute([$input['name'],$input['category'],$input['price'],$input['stock']]);
}
echo json_encode(['ok'=>true]);
```

## Examen Interactivo de PLP III: La Aventura del Desarrollador Web 🎮

### Frontend

#### index.php

```
<?php /* AR - FoodExpress */ ?>
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>AR FoodExpress</title>
  <link rel="stylesheet" href="css/AR_estilos.css" />
</head>
<body>
<header class="ar-header">
  <h1 class="ar-logo">TT FoodExpress</h1>
  <nav class="ar-nav">
    <button data-cat="all">Todos</button>
    <button data-cat="Pizzas">Pizzas</button>
    <button data-cat="Burgers">Burgers</button>
    <button data-cat="Bebidas">Bebidas</button>
  </nav>
  <div class="ar-cart-badge" id="arCartBadge">0</div>
</header>

<main class="ar-grid">
  <section>
    <h2>Menú</h2>
    <div id="arProducts" class="ar-products"></div>
  </section>
  <aside>
    <h2>Carrito</h2>
    <div id="arCart"></div>
    <div class="ar-totals">Subtotal: $<span id="arSubtotal">0</span></div>
    <form id="arCheckout">
      <input required name="name" placeholder="Nombre" />
      <input required name="phone" placeholder="Teléfono" />
      <input required name="address" placeholder="Dirección" />
      <button type="submit">Confirmar Pedido</button>
    </form>
    <div id="arMsg"></div>
  </aside>
</main>

<script src="js/AR_script.js"></script>
</body>
</html>
```

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

js/AR\_script.js

```
/* AR_script.js
- Lista productos por categoría (fetch a /api/products_list.php)
- Carrito en memoria con add/remove/update y subtotal automático
- Badge con cantidad total
- Checkout: POST JSON a /api/order_create.php y muestra confirmación
*/
const $ = (q)=>document.querySelector(q);
const $$ = (q)=>document.querySelectorAll(q);

const state = { products: [], cart: [] };

function renderProducts() {
  const c = $("#ttProducts");
  c.innerHTML = state.products.map(p => `
    <article class="tt-card">
      <h3>${p.name}</h3>
      <small>${p.category}</small>
      <div>$$${p.price.toFixed(2)}</div>
      <button data-id="${p.id}" data-price="${p.price}" data-
name="${p.name}">Agregar</button>
    </article>
  `).join("");
  c.onclick = (e)=>{
    const b = e.target.closest("button[data-id]");
    if (!b) return;
    addToCart({ id:+b.dataset.id, name:b.dataset.name, price:+b.dataset.price, qty:1 });
  };
}

function addToCart(item) {
  const i = state.cart.findIndex(x=>x.id===item.id);
  if (i>=0) state.cart[i].qty++;
  else state.cart.push(item);
  renderCart();
}

function removeFromCart(id) {
  state.cart = state.cart.filter(x=>x.id!==id);
  renderCart();
}

function updateQty(id, qty) {
  qty = Math.max(1, qty|0);
  const it = state.cart.find(x=>x.id===id);
  if (it) it.qty = qty;
  renderCart();
}
```

## Examen Interactivo de PLP III: La Aventura del Desarrollador Web 🎮

```
function renderCart() {
  const c = $("#ttCart");
  if (!state.cart.length) { c.innerHTML = "<em>Vacío</em>"; }
  else {
    c.innerHTML = state.cart.map(it => `
      <div class="tt-row">
        <span>${it.name}</span>
        <input type="number" min="1" value="${it.qty}" data-id="${it.id}" />
        <span>${(it.qty*it.price).toFixed(2)}</span>
        <button class="tt-remove" data-id="${it.id}">X</button>
      </div>
    `).join("");
  }
  // Eventos
  c.oninput = (e)=>{
    const inp = e.target.closest('input[type="number"]');
    if (inp) updateQty(+inp.dataset.id, +inp.value);
  };
  c.onclick = (e)=>{
    const btn = e.target.closest('.tt-remove');
    if (btn) removeFromCart(+btn.dataset.id);
  };

  // Subtotal y badge
  const subtotal = state.cart.reduce((s,it)=>s+it.qty*it.price,0);
  $("#ttSubtotal").textContent = subtotal.toFixed(2);
  $("#ttCartBadge").textContent = state.cart.reduce((s,it)=>s+it.qty,0);
}

async function loadProducts(category='all') {
  const res = await
  fetch(`api/products_list.php?category=${encodeURIComponent(category)}`);
  state.products = await res.json();
  renderProducts();
}

async function checkout(e) {
  e.preventDefault();
  if (!state.cart.length) return $("#ttMsg").textContent = "Agrega productos al carrito";
  const fd = new FormData(e.target);
  const customer = Object.fromEntries(fd.entries());
  const body = JSON.stringify({ items: state.cart, customer });
  const res = await fetch('api/order_create.php',{ method:'POST', headers:{'Content-Type':'application/json'}, body });
  const data = await res.json();
  if (data.ok) {
    $("#ttMsg").textContent = `Pedido #${data.order_id} creado. Total
    ${data.total.toFixed(2)}`;
    state.cart = [];
  }
}
```



### Examen Interactivo de PLP III: La Aventura del Desarrollador Web 🎮

```
renderCart();
e.target.reset();
} else {
  $("#ttMsg").textContent = "Error creando el pedido";
}
}

window.addEventListener('DOMContentLoaded', ()=>{
  $$('#\\.tt-nav button, .tt-nav
button').forEach(b=>b.addEventListener('click',()=>loadProducts(b.dataset.cat)));
  $("#ttCheckout").addEventListener('submit', checkout);
  loadProducts('all');
});
```

#### AR\_estilos.css

```
:root { --gap: 1rem; font-family: system-ui, sans-serif; }
* { box-sizing: border-box; }
body { margin: 0; background:#0b0b0b; color:#f7f7f7; }
.tt-header { display:flex; gap:var(--gap); align-items:center; padding:var(--gap);
background:#161616; position:sticky; top:0; }
.tt-nav button { margin-right:.5rem; }
.tt-cart-badge { margin-left:auto; background:#fff; color:#000; padding:.25rem .5rem;
border-radius:999px; min-width:2rem; text-align:center; }
.tt-grid { display:grid; gap:var(--gap); grid-template-columns: 2fr 1fr; padding:var(--gap); }
.tt-products { display:grid; grid-template-columns: repeat(auto-fill, minmax(180px,1fr));
gap:var(--gap); }
.tt-card { background:#1e1e1e; padding:var(--gap); border-radius:.75rem; }
.tt-row { display:grid; grid-template-columns: 1fr 80px 100px 40px; gap:.5rem; align-
items:center; margin:.25rem 0; }
@media (max-width: 900px){ .tt-grid { grid-template-columns: 1fr; } }
button { cursor:pointer; }
button:focus, input:focus { outline: 2px solid #888; outline-offset: 2px; }
```

## **Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

### **📋 PROYECTO C: "QuizMaster" - Plataforma de Trivia**

*Una institución educativa quiere gamificar el aprendizaje*

#### **Requisitos Funcionales:**

- Sistema de preguntas con múltiple opción (mínimo 15 preguntas)
- Validación de respuestas en tiempo real
- Sistema de puntuación y temporizador
- Tabla de mejores puntajes (stored en BD)
- Categorías temáticas con dificultad variable

#### **Requisitos No Funcionales:**

- Mínimo 3 secciones (inicio, juego, resultados)
- Animaciones fluidas y feedback inmediato
- Diseño responsive
- Interfaz intuitiva sin instrucciones complejas

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

⚡ **NIVEL 2: Interactividad con JavaScript (20 puntos)**

📄 **Documenta:** Comenta la funcionalidad al inicio del archivo JS (3-5 líneas).

**Para PROYECTO A (MusicStream):**

Implementar: Reproductor Interactivo con Playlist

- Play/Pause/Skip con controles visuales
- Barra de progreso funcional
- Lista de reproducción dinámica
- Almacenar última canción reproducida

**Para PROYECTO B (FoodExpress):**

Implementar: Carrito de Compras Dinámico

- Agregar/eliminar productos sin recargar
- Cálculo automático de subtotales
- Validación de cantidades
- Mostrar contador de items en el carrito

**Para PROYECTO C (QuizMaster):**

Implementar: Lógica de Juego Completa

- Algoritmo de validación de respuestas
- Sistema de puntuación progresiva
- Temporizador con penalización
- Feedback visual inmediato (correcto/incorrecto)

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

🔑 **NIVEL 3: Backend con PHP (20 puntos)**

⚠️ **OBLIGATORIO:** Conexión e interacción con Base de Datos MySQL

📝 **Documenta:** Comenta la funcionalidad PHP implementada.

Implementación Requerida:

**Para MusicStream:**

- CRUD de canciones/álbumes
- Sistema de favoritos persistente
- Búsqueda con queries SQL

**Para FoodExpress:**

- CRUD de productos
- Registro de pedidos en BD
- Cálculo de totales en servidor

**Para QuizMaster:**

- Banco de preguntas desde BD
- Sistema de ranking persistente
- Registro de partidas jugadas

📊 **Base de Datos:**

**Crear con mínimo 2 tablas relacionadas:**

- Claves primarias y foráneas
- Datos de prueba (mínimo 10 registros)

**Exportar:**

- **[iniciales]\_estructura.sql**
- **[iniciales]\_datos.sql**

**Examen Interactivo de PLP III: La Aventura del Desarrollador Web** 🎮

🧠 **NIVEL 4: Diseño y Experiencia Visual (20 puntos)**

📄 **Documenta:** Explica tus decisiones de diseño (paleta, tipografía, layout).

**Requisitos Funcionales:**

- Paleta de colores coherente (4-5 colores)
- Tipografía consistente (jerarquía clara)
- Responsive: 3 breakpoints mínimo
- Menú adaptativo (hamburguesa en mobile)

**Requisitos No Funcionales:**

- Transiciones suaves (hover, focus)
- Loading states visibles
- Contraste adecuado (accesibilidad)
- Espaciado uniforme (grid/flexbox)

## Examen Interactivo de PLP III: La Aventura del Desarrollador Web 🎮

### 📦 ENTREGA FINAL

#### Estructura del Proyecto:

```
[INICIALES]_Parcial_PLP3/
├── index.html (o index.php)
├── css/
│   └── [iniciales]_estilos.css
├── js/
│   └── [iniciales]_script.js
├── includes/
│   └── [iniciales]_conexion.php
├── [iniciales]_assets/
│   └── images/
├── database/
│   ├── [iniciales]_estructura.sql
│   └── [iniciales]_datos.sql
├── docs/
│   └── [APELLIDO]_[NOMBRE]_Parcial.pdf
└── README.md
```

#### 📁 Método de Entrega:

1. Archivo ZIP: [\[APELLIDO\]\\_\[NOMBRE\]\\_PLP3.zip](#)
2. Repositorio GIT con commits descriptivos
3. Subir a aula virtual dentro del tiempo del examen

### 🏆 EVALUACIÓN

#### Puntos Bonus (+10 máximo):

- ✨ Creatividad excepcional (+3)
- 🔒 Seguridad (prepared statements) (+2)
- ♿ Accesibilidad (ARIA, semántica) (+2)
- 🚀 Features avanzadas (+3)

## Examen Interactivo de PLP III: La Aventura del Desarrollador Web 🎮

### Penalizaciones:

- ❌ Sin nomenclatura de prefijos: -5 pts
- ❌ Código sin comentarios: -3 pts
- ❌ No funciona: -10 pts
- ❌ Plagio: 0 en el parcial

### 🎯 CHECKLIST FINAL

- Teoría completa
- Nomenclatura con prefijo
- Proyecto funcional
- BD exportada
- CSS responsive
- JavaScript comentado
- PHP con BD funcionando
- README.md claro
- GIT con commits
- ZIP correctamente nombrado

🚀 ¡ÉXITO, DESARROLLADOR!

👉 ¡Que la fuerza del código esté contigo! 🍵