Universidad de la Cuenca del Plata
Autorización Definitiva. Decreto Poder Ejecutivo Nacional N°091/2006

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

Ingeniería en Sistemas de Información	
Cátedra: Paradigmas y lenguajes de programación III	**Profesor**: Mgter. Ing. Agustín Encina
Alumno: Becerra Tobias	**Fecha**: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

Respuesta:

El modelo cliente-servidor esta basado en peticiones (request) y respuestas (responses). Un cliente solicita un recurso a un servidor y este procesa la peticion, donde finalmente devuelve una respuesta que es el recurso solicitado.

Esta compuesto por:

- Cliente (Navegador)
- Servidor (Web Server)
- HTTP (Protocolo de Transferencia de Hipertexto)
- URL (Localizador Uniforme de Recursos)

El ciclo de vida de este se basa en:

Petición: El usuario ingresa una URL y el navegador genera una petición HTTP, la cual envia al servidor correspondiente

Procesamiento: El servidor recibe esta petición y localiza el recurso o ejecuta un script para generar la respuesta.

Respuesta: El servidor envia una respuesta HTTP que tiene un código de estado, ya sea 200, 300, 400, 500 , etc y el contenido.

Un ejemplo de un caso real podriamos tomar en cuenta la pagina de la facultad, donde uno quiere ver sus inasistencias por ejemplo y solicitando el boton de “Inasistencias” este nos devuelve una tabla con la informacion requerida.

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

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

Respuesta:

La diferencia principal entre estos es que los selectores de clase aplican a todos los elementos con class = “mi-clase”, en cambio, el selector de ID aplica al único elemento con id= “mi-id”

```
//EJEMPLO 1

<section id="comidas-list" class="section">

// En CSS

#comidas-list {

        gap: 10px;

        padding: 5px;

        width: calc(100% - 10px);

    }

// EJEMPLO 2

<div id="carrito-vacio" style="display:none;">

                <p>Tu carrito está vacío</p>

                <a href="/TAB_PARCIAL_PLP3/" class="btn-volver">Volver
al menú</a>

            </div>
```

```
#carrito-vacio {

        padding: 20px 10px;

    }
```

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

Respuesta

Las variables en JS son un valor que se guarda en un espacio de memoria y se utiliza para representar datos o conjuntos de datos.

La diferencia que existe entre var, let y const es que var solamente tiene el alcance de una funcion misma, mientras que let y const tienen un alcance global.

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)

Respuesta

PHP es un lenguaje de programacion server-side, es decir, que solamente se ejecuta en el servidor web y no en el navegador del usuario. Una diferencia principal con los lenguajes

front end es que justamente no se ejecuta en el navegador del usuario y por lo tanto, no se puede ver a simple vista lo que realice.

El proposito como tal de php es crear paginas web dinamicas e interactuar con bases de datos.

El ciclo de vida de PHP funciona de la siguiente forma:

Petición: Un usuario en su navegador (cliente) solicita una página, por ejemplo, index.php.

Identificación: El servidor web (como Apache o Nginx) tiene la extensión .php y sabe que no debe enviar el archivo directamente.

Procesamiento: El servidor pasa el archivo al intérprete de PHP .

Ejecución: El intérprete PHP lee el archivo, ejecuta solo el código PHP (se conecta a una base de datos, obtiene el nombre del usuario, procesa un formulario, etc.).

Respuesta: El intérprete genera una salida de texto plano , que casi siempre es HTML .

Envío: El servidor web toma ese HTML (ya sin código PHP) y se lo envía al navegador del usuario.

Codigo de ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>

  <h1>Lista de Tareas</h1>
  <ul>
    <li>Tarea 1</li>
    <li>Tarea 2</li>

    <?php
      // Esto es código del servidor
      $usuario = "Juan Pérez";
      echo "<li>Tarea especial para: $usuario</li>";
    ?>
    <li>Tarea 3</li>
  </ul>
```

```
</body>
</html>
```

🚀 *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?

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?

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

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:

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

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

🎨 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

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! 🎃