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:Renzo Gómez Terrusi	**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 🎮

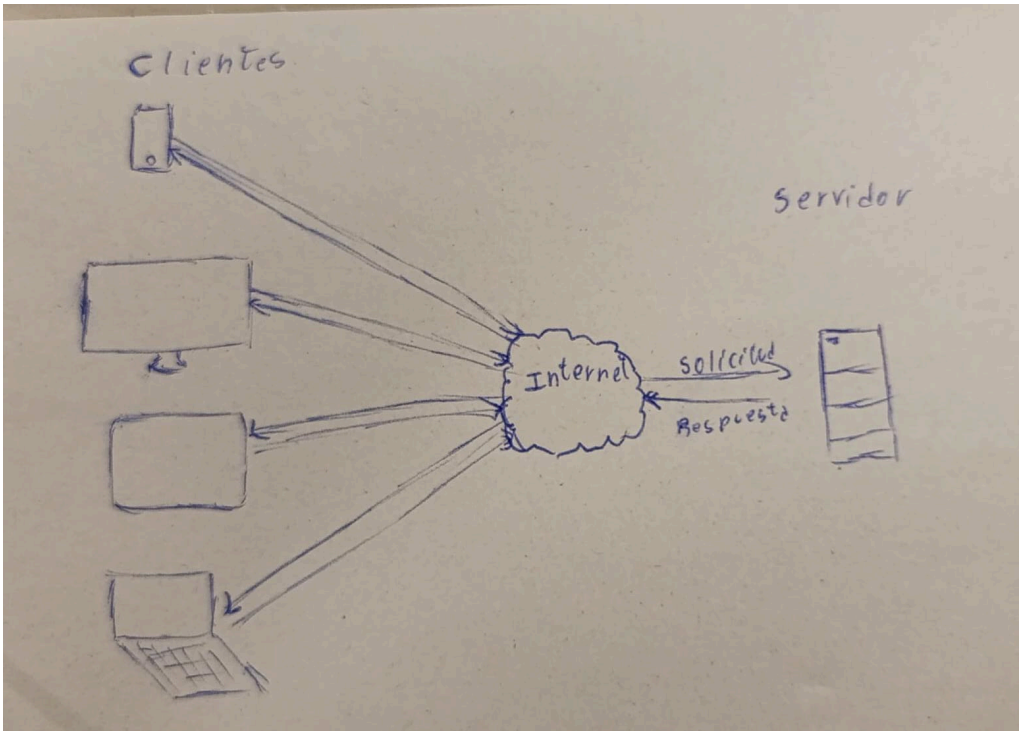
🌐 *RUTA A: "El Arquitecto Web" (Fundamentos y Estructura)*

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



La arquitectura cliente servidor se refiere a la arquitectura mediante la cual todos los sitios web del mundo actualmente se comunican con el usuario final. Se trata de un modelo que consta de 2 componentes: un cliente (el usuario que solicita servicios), y el servidor, (un dispositivo que proporciona servicios solicitados por el usuario). El flujo de comunicación es el siguiente: El cliente hace una solicitud mediante el protocolo HTTP ó HTTPS (Hiper Text Transfer Protocol Secure) a el servidor, el servidor la procesa y devuelve una respuesta en este mismo protocolo. De esta forma se establece un modelo de comunicación mediante peticiones y respuestas.
Un ejemplo de esto podría ser el campus virtual, en el cual los clientes (alumnos) solicitan al servidor de la UCP información como sus notas, trabajos, cátedras, etc., luego el servidor

procesa dichas solicitudes y en caso de cumplir condiciones como credenciales correctas envía una respuesta con esos datos

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

Los selectores nos permiten elegir elementos específicos de nuestro css para asegurarnos de que aunque tengamos, por ejemplo dos divs, podamos diferenciarlos claramente y podamos asignarle estilos individuales.

Los selectores de clase nos permiten elegir a todos los elementos que sean parte de una clase específica, independientemente incluso de que sean del mismo tipo de elemento (mientras los atributos que les queramos cambiar sean comunes entre distintos elementos) como vemos a continuación

```
<p class="note">
```

```
 <li class="note" value="2">

.note{

  font-size: 20px;

}
```

Por otro lado, los selectores de ID apuntarán a un único elemento. Nos sirven si queremos asegurarnos de que algún cambio se efectúe a un único elemento

```
<li class="note" id="id-selector-demo" value="3">

#id-selector-demo{

  color: green;}
```

En un código corriendo ambos css el elemento con ID recibirá los cambios efectuados por su selector ID prioritariamente antes que los de su clase dada la jerarquía de especificidad de css.

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

Las variables en JavaScript sirven para almacenar datos que sean relevantes para la ejecución de las funciones y operaciones de nuestra página. De esta forma podemos reutilizar o iterar sobre estos datos de forma fácil.

Var: Sirve para declarar variables “globales”, es decir, variables que se ejecutan fuera de un contexto específico.

Let: Sirve para declarar variables dentro de un contexto específico, como el de una función.

Const: Sirve para almacenar datos que no van a cambiar durante la ejecución de nuestra página.

Hoisting: Es la funcionalidad de JS que hace que a la hora de ejecutar el código las declaraciones de las variables var automáticamente se ejecuten primero. De tal forma que incluso si declaramos una variable después de usarla nuestro código seguirá funcionando. Esto sin embargo no ocurren con otras variables como let o const.

Scope: Nos indica el alcance de una variable. Tenemos Global Scope, que abarca todo el código; Function Scope, que abarca una función específica; y Block Scope, que abarca un bloque de código específico definido entre corchetes.

```
{
  let Ejemplo = “Buenas Tardes”;
}
{
  var ejemplo = “Porfa apruebeme :”;
```

```
}
```

Fuera de esos dos bloques de código únicamente el declarado con var será accesible

```
Hoisting = “Funciona”;

elem = document.getElementById(“prueba");

elem.innerHTML = Hoisting;

var Hoisting;
```

La variable podrá ser accedida por el get.innerHTML dado que gracias al hoisting su declaración se pasa al inicio.

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)

PHP Hipertext Preprocessor es un lenguaje de alto nivel que es embebido en el HTML de nuestra página con fin de hacer que la misma tenga funcionalidades más complejas que simplemente tener un buen Frontend. Se encarga del Backend o de las funciones y manejo de la base de datos de la página.
Su principal diferencia con el frontend es que en sí mismo no está diseñado como para mostrarle algo directamente al usuario, está diseñada para agregar funcionalidades que este no necesariamente va a interpretar en php, sino en html mediante inyecciones del mismo con funcionalidades como echo.
El procesamiento de un formulario se realiza de la siguiente manera

```
<main class="content">
    <form method="post" action="formulario.php">
                Ingrese su Nombre: <input type="text" name="nombre"
required><br>
        Ingrese su Edad: <input type="number" name="edad" required><br>
        <input type="submit" value="Enviar">
    </form>
</main>
```

```
formulario.php
<?php
        echo "El nombre ingresado es: {$_REQUEST['nombre']}";
        echo "<br>";
        echo "La edad ingresada es: {$_REQUEST['edad']}";
    ?>
```

🚀 *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)

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

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

🎨 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

Hacer esto con Renzo Gomez Terrussi

🎯 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! 🎃