

Ingeniería en Sistemas de Información					
Cátedra: programac		у	lenguajes	de	Profesor: Mgter. Ing. Agustín Encina
Alumno:				Fecha:	

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! 🎯

X 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:

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





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

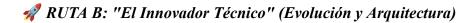
Desafío 4 - Introducción a PHP (5 puntos)

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)





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:

- 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)

III 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

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



□ PROYECTO A: "MusicStream" - Plataforma de Música Online

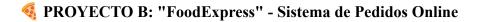
Una discográfica indie quiere su propia plataforma de streaming

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)

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





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

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





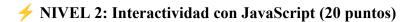
Una institución educativa quiere gamificar el aprendizaje

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

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





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)



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





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)

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





Estructura del Proyecto:

```
| INICIALES]_Parcial_PLP3/
| — index.html (o index.php)
| — css/
| — [iniciales]_estilos.css
| — js/
| — includes/
| — 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

Y EVALUACIÓN

Puntos Bonus (+10 máximo):

- Creatividad excepcional (+3)
- Seguridad (prepared statements) (+2)
- 6 Accesibilidad (ARIA, semántica) (+2)
- Features avanzadas (+3)



Penalizaciones:

- X 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! 🎃