

Prueba Técnica - Desarrollador Full Stack (PHP, JavaScript, jQuery, SQL)

Duración estimada: 2 a 3 horas

Modalidad: Asincrónica (sin supervisión)

Nivel del cargo: Full Stack

Objetivo: Evaluar lógica, buenas prácticas y capacidad para construir soluciones robustas.

Reglas:

Debes publicar las respuestas en tu Git y enviarnos un enlace para conocerlas. No se recibirán respuestas al correo.

Plazo máximo de publicación y entrega es: 2025-08-12 11:59 am.

Los cambios o ajustes publicados después de las 12 del día del 12 de agosto no serán tenidos en cuenta.

Cada sección Debería tener una carpeta y cada punto un archivo de respuesta en el Proyecto.

Nota: Sabemos que en la actualidad puedes usar herramientas LLM para dar respuesta directa a cada pregunta, te solicitamos el favor NO hacer uso de las mismas por lo pronto en esta prueba.

Sección 1 - Conocimientos Técnicos (15 puntos)

Responde brevemente las siguientes preguntas:

1. ¿Cuál es la diferencia entre '==' y '===' en PHP?

R/= La diferencia es el nivel de rigor por la cual se van a evaluar dichos valores donde el == compara solo el valor de dichas variables, en cambio el === se tiene en cuenta tanto el valor como el tipo de dato.

2. ¿Qué valor retorna 'NaN == NaN' en JavaScript y por qué?

R/= Esto es debido a que NaN es una representación numérica que a su vez no es un número válido, esto quiere decir que nunca serán iguales.

3. ¿Qué hace .on('click', ...) comparado con .click() en jQuery?

R/= El primero el elemento debe existir ya en el DOM, el segundo son aquellos listeners que funcionan en formularios dinámicos y pueda que dicho elemento no exista todavía en el DOM, es decir, queda a la escucha hasta que dicho elemento sea creado dinámicamente en el DOM.

4. ¿Qué diferencia hay entre INNER JOIN y LEFT JOIN en SQL?

R/= INNER JOIN Retorna cuando hay coincidencia en las tablas involucradas en el select y el LEFT JOIN retornaría todos los datos de la tabla de la izquierda así no coincidan con el de la izquierda, es decir, que los datos de la tabla de la derecha aparecerían en NULL.

5. ¿Qué es una condición falsy? Da tres ejemplos en PHP y JavaScript.

R/= Son aquellas condiciones que evalúan una variable booleana que vaya a ser considerada falsa aunque realmente no sea falsa

Ejemplos mas comunes son

If(null) – if(0) – if(Nan) – if(false)

```
$valores = [false, 0, "", "", NULL, []];  
foreach ($valores as $v) {  
    var_dump((bool)$v);  
}
```

Sección 2 - Lógica de Programación (40 puntos)

Desarrolla las siguientes funciones en el lenguaje que prefieras (PHP o JavaScript). Escribe código limpio y comenta si lo consideras necesario.

1. Palabra más larga: Dada una cadena, encuentra la palabra más larga.
2. Paréntesis balanceados: Implementa una función que retorne true si los paréntesis están balanceados.
3. Frecuencia de caracteres: Crea una función que cuente la frecuencia de cada letra en una cadena.
4. FizzBuzz extendido: Imprime los números del 1 al 100 según la regla Fizz, Buzz, FizzBuzz (y solo positivos).
5. Define un modelo de datos donde podamos registrar información de visitas que realizamos a diferentes puntos de gestión y luego podamos construir un modelo analítico para poder visualizar que tan lejos esta un punto de otro y cuáles son los puntos que mayor cantidad de visitas se han ejecutado, cuáles son los puntos donde más se demoran dichas visitas.

Sección 3 - Desarrollo Práctico (35 puntos)

- Ejercicio 1 – Construcción de API REST en PHP (20 puntos)

Construye una API RESTful en PHP (sin frameworks) que exponga un endpoint GET `/usuarios/recientes` y devuelva los usuarios registrados en los últimos 30 días desde la base de datos `usuarios` (campos: id, nombre, email, fecha_registro). La respuesta debe estar en formato JSON. Incluye manejo de errores, buenas prácticas y documentación en comentarios.

- Ejercicio 2 – Interacción DOM con JavaScript y jQuery (15 puntos)

Crea una página con un campo de texto (id='nombre'), un botón (id='btnGuardar') y un div (id='resultado'). Al hacer clic en el botón, muestra en el div el contenido del campo convertido a

mayúsculas.

Utiliza jQuery para el manejo de eventos y vanilla JS para la lógica.

Preguntas Abiertas (10 puntos)

¿Qué consideras una buena práctica que aplicas siempre en tus desarrollos y por qué?

R/= En mi caso siempre utilizar extensiones en mi VSC como elephant para saber que código puedo suprimir, indicarme que variables uso y que no, para no dejar variables, funciones o métodos expuestos para un ataque, en el caso de consultas SQL usar PDO para evitar inyecciones SQL, en los API REST siempre es buena práctica manejar JWT para el uso de otros endpoints.

Si el desarrollo es nativo siempre manejar MVC y dentro de cada una manejar carpetas por módulos, si es en framework, ya dependiendo del framework establezco reglas por módulos para una mejor escalabilidad.

Cuando el código es muy repetitivo, lo que procuro es crear un método dentro de un archivo global para poder ser reutilizado en varios métodos las veces que sea necesario y no estar repitiendo mucho código.

Ya de costumbre realizo pruebas unitarias y escaneos de mi código para evitar vulnerabilidades. Cuando son métodos que se desarrollan con bodyrequest, procuro a generar métodos para verificar el request, sus valores (sanitizar valores).

Siempre procuro, para no generar una función muy larga, refactorizo el código para una mejor escalabilidad en caso de soportes.

¿Cuándo y por qué usar GIT y GitHub?

En Git es bueno ya que me permite tener un control de versiones ya que si al implementar un desarrollo o una mejora de algo y sale malo, simplemente vuelve a un punto con un checkout y el id del commit.

GitHub es demasiado importante hoy en día por el tema del CI/CD, para despliegues continuos, también para el tema de compartir repositorios con todo el equipo de trabajo, generar secretos para no exponer información importante. Hoy en día GitHub inclusive cuenta con muchas herramienta que al hacer un push o un merge, el GitHub al detectar un cambio, realiza escaneos del código para evitar vulnerabilidades, errores o problemas de seguridad.

¿En la Informática y en específico en el Desarrollo de Software que es lo más importante antes de Iniciar con el Desarrollo. ?

Problemática

Objetivos del proyecto

Alcance del proyecto

Requerimientos Funcionales y no Funcionales

Listado de Actividades con responsabilidades y tiempos

Listado y análisis de stakeholders

Definición de reglas de negocio

Documentación con arquitectura y diagramas UML

Diseño UX (mockups)

¿Si te pido realizar algo fuera de la expectativa de tu cargo, como lo tomas? ¿Si no sabes lo que te solicito es motivo de discusión o aprendizaje, como lo abortas paso a paso?

R/= Si se me pide realizar algo fuera de mi cargo y tengo las condiciones, lo asumiré con la mejor actitud y compromiso. En caso de que no sepa cómo hacerlo, lo tomo como aprendizaje ya que me permite crecer profesionalmente y aportar más valor a la empresa.