

Test Autoevaluación



Prueba entregada



vare

- Fecha de entrega No hay fecha de entrega
- Puntos 10
- Preguntas 10
- Disponible 9 de sep a las 0:00 - 29 de jun de 2025 a las 23:59
- Límite de tiempo Ninguno
- Intentos permitidos Ilimitado

Instrucciones

Test de autoevaluación



Este test incluye un total de **21 preguntas** sobre los diferentes contenidos aprendidos en esta unidad.

¡Adelante!

[Volver a hacer la prueba](#)

Historial de intentos

	Intento	Hora	Puntuación
MÁS RECIENTE	Intento 1	2 minutos	7 de 10

Puntuación para este intento: 7 de 10

Entregado 19 de nov a las 12:10

Este intento ha tenido una duración de 2 minutos.



Pregunta 1

1 / 1 puntos

¿Cuál es la complejidad del algoritmo *Insertion Sort*?

- ☐ $O(n)$ en el mejor caso y $O(n)$ en el peor caso.
- ☐ $O(n^2)$ en todo caso.

¡Correcto!

- ☒ $O(n)$ en el mejor caso y $O(n^2)$ en el peor caso.


¡Exacto! El mejor caso es una lista ya ordenada, y el peor, una ordenada al contrario.

- ☐ $O(n \cdot \log(n))$ en todo caso.



Pregunta 2

1 / 1 puntos

 Prueba entregada

Un lenguaje orientado a objetos

- ☐ Nunca puede pasar funciones como argumentos o valores de retorno.
- ☐ Nunca es de tipo procedimental.
- ☐ Es un tipo de lenguaje declarativo basado en el concepto de 'Objetos' y 'Clases'.

¡Correcto!

- ☒ Es un tipo de lenguaje imperativo basado en el concepto de 'Objetos' y 'Clases'.

¡Exacto! Es un paradigma dentro de los lenguajes imperativos en el que los Objetos son entidades que almacenan datos y estado.



Pregunta 3

0 / 1 puntos

El Test Doubling es una técnica que permite:

- ☐ Parametrizar tests de forma sencilla.
- ☐ Invocar tests en servidores remotos.

Respuesta correcta

- ☐ Simular servicios reales con código fuente.

Respondido

- ☒ Sustituir llamadas a servicios reales por otras.



Pregunta 4

1 / 1 puntos

¿Cuál de las siguientes afirmaciones sobre TDD es válida?

- ☐ Significa *Test Delimited Domains* y delimita qué se debe probar y qué no.
- ☐ Significa *Test Delimited Development* y delimita qué se debe probar y qué no.
- ☐ Significa *Test-Driven Development* y delimita qué se debe probar y qué no.

¡Correcto!

- ☒ Significa *Test-Driven Development* y se basa en escribir los tests antes que el código productivo.



Pregunta 5

0,5 / 1 puntos

Cuando creamos un script para una tarea concreta:

¡Correcto!

- ☒ Siempre debemos usar buenas prácticas de codificación y escribir tests.

No se debe hacer siempre, lo fundamental es que hagan bien su trabajo.

- ☐ Siempre lo podremos ejecutar en cualquier sistema.

Respuesta correcta

- ☐ Lo almacenaremos en un repositorio distinto al código al que da soporte
- ☐ Siempre se escribe usando Bash.



Pregunta 6

1 / 1 puntos

✓ Prueba entregada

¿Cuál es la complejidad del algoritmo *Selection Sort*?

☐ $O(n)$ en el mejor caso y $O(n^2)$ en el peor caso.

¡Correcto!

☒ $O(n^2)$ en todo caso.

¡Exacto! No existe mejor o peor caso, siempre se recorre la lista una vez por elemento: $n + (n-1) + (n-2) + \dots$

☐ $O(1)$ en el mejor caso y $O(n)$ en el peor caso.

☐ $O(n \cdot \log(n))$ en todo caso.



Pregunta 7

1 / 1 puntos

Un test que comprueba que mi aplicación se conecta correctamente a una base de datos MySQL es:

☐ Un test unitario.

☐ Un test end-to-end.

¡Correcto!

☒ Un test de integración.

¡Exacto! Podría ser un test funcional también, pero es el tipo de integración que interesa poder probar con Mock Servers o incluso con contenedores.

☐ Un smoke test.



Pregunta 8

1 / 1 puntos

¿Qué hace el siguiente bloque de código funcional en Java 8?

```
new Random().ints().limit(100).filter( i -> i < 30 ).map( i -> i^2 ).sum();
```

☐ Calcula la suma de los cuadrados de 100 enteros aleatorios.

☐ Filtra 100 valores aleatorios y calcula la suma de sus cuadrados.

☐ Calcula la suma de los cuadrados de 100 enteros aleatorios menores de 30.

¡Correcto!

☒ Filtra 100 números enteros aleatorios y calcula la suma de los cuadrados de los menores de 30.

¡Exacto!



Pregunta 9

0 / 1 puntos

¿Qué hace el siguiente bloque de código funcional en JavaScript?

```
Array.from({
  Math.random()
}).filter( x =>
  x < 30
).reduce( (a,b) => a*a + b*b )
```

✓ Prueba entregada

- ☐ Calcula la suma de los cuadrados de 100 números aleatorios.

Respuesta correcta

- ☐ Filtra 100 números aleatorios y calcula la suma de sus cuadrados.
- ☐ Calcula la suma de los cuadrados de 100 números aleatorios menores de 30.

Respondido

- ☒ Filtra 100 números enteros aleatorios y calcula la suma de los cuadrados de los menores de 30.

No, no son números enteros.



Pregunta 10

0,5 / 1 puntos

¿Cuál de estas afirmaciones es cierta?

- ☐ Código fácil de entender es código que no ha costado nada escribir.
- ☐ Escribir código más claro siempre implica mayor calidad del software.

Respuesta correcta

- ☐ Tardar más tiempo en escribir código implica mayores costes o pérdida de alcance del software.

¡Correcto!

- ☒ Código fácil de entender es código fácil de mantener.

Está afirmación es correcta.

Puntuación de la prueba: 7 de 10