

Prueba técnica proceso selección estancia en prácticas 2025.

El objetivo de esta prueba es valorar los conocimientos básicos de programación orientada a objetos que tienes. Puedes utilizar la web o cualquier modelo de IA para ayudarte, pero ten en cuenta que debes comprender tu código a la perfección porque podríamos solicitarte que explicaras cualquier clase y/o método en las entrevistas individuales. Lo ideal sería que desarrollaras el código íntegramente tú, así podrías explicarlo con facilidad.

Para la entrega deberás, idealmente, crear un repositorio en tu cuenta de GitHub donde subirás todos los archivos generados. El enlace a tu repositorio (que tiene que ser público) deberás mandarlo por mail a esta cuenta: jano.soler@cuatroochenta.com. Si esto no es posible, crea una carpeta con todos los archivos, comprímela y mándala a esa misma dirección.

Vas a crear dos archivos para sendas clases en el lenguaje de programación que prefieras. Uno para una clase llamada Cd y otro para la clase Disco duro. Debes pensar si esas clases deberían heredar de alguna otra clase o implementar cualquier interfaz (si no has programado nunca con interfaces, puedes utilizar herencia únicamente).

Para ello, antes de implementar nada, debes considerar qué código colocar en la o las interfaces en caso de existir y lo mismo en caso de crear alguna clase de la que heredar.

A continuación, se muestran algunos comportamientos que comparten tanto los CD como los discos duros:

- Ambos giran.
- Ambos almacenan datos.
- Los datos se escriben en los discos mediante el uso de un láser (CD) o por un cabezal magnético (disco duro).
- Los datos de los discos se pueden leer utilizando un láser (CD) o por un cabezal magnético (disco duro).
- Una vez introducidos, ambos reportan información como nombre, capacidad, contenido y tipo de disco.

Añade más comportamientos a la lista si los consideras relevantes. Ten en cuenta que la información como la capacidad de almacenamiento NO pertenece a esta lista, ya que describe una propiedad (campo) en lugar de un comportamiento (método).

Para cada comportamiento, identificad si depende de algún tipo de atributo de instancia. Por ejemplo, el comportamiento de giro no requiere ninguna información de campo de las clases de CD o Disco duro, en cambio reportar la información de cada disco sí.

Esta recopilación y clasificación de comportamientos (métodos) es el primer paso hacia el diseño de interfaces o clases heredadas. Al hacerlo tienes que considerar qué comportamientos se pueden aplicar generalmente frente a cuáles dependen de los datos almacenados en una instancia específica de un objeto. Una vez tengas esto claro, debes decidir qué comportamientos entrarían en una hipotética interfaz o interfaces o en clases heredadas.

Los comportamientos añadidos a alguna modalidad de herencia NO tienen que ser idénticos para cada objeto que herede o implemente la interfaz. Por ejemplo, aunque tanto los CD como los discos duros giran, lo hacen a diferentes velocidades. Su comportamiento, girar en este caso, es el mismo, pero los detalles de ese comportamiento varían. Por tanto, un método como *girarDisco()* encaja bien en una interfaz o una clase madre.

Para aquellos métodos que hayas decidido incluir en alguna interfaz o forma de herencia, piensa en las diferencias en estos comportamientos entre los CD y los discos duros.

La lógica de los métodos será muy sencilla, por ejemplo, el método girar de un CD lo único que haría sería devolver un mensaje por consola tipo: “*CD girando...*” y el resto de los métodos deberían ser similares. Simplemente devolver un mensaje que los diferencie.

También debes escribir los campos necesarios para definir las propiedades de un CD o disco duro (por ejemplo, capacidad de almacenamiento).

Considera:

- ¿Qué campos comparten ambos tipos de discos y cuáles son exclusivos de cada clase?.
 - ¿Alguno de los campos representan constantes?
- Deberás tener en cuenta también la implementación de todos los getters y setters necesarios para gestionar los atributos.

Cuando todo este proceso haya acabado, cread un archivo que importe ambas clases y declarad e inicializad un objeto de CD y otro disco duro, luego ejecutad cada uno de los comportamientos que hayáis implementado.

BONUS:

Los CD y discos duros no son los únicos medios planos y circulares que se han utilizado para almacenar y devolver datos.

- Añade una clase para discos de vinilo (LP) y otra para blueRay.
- ¿Si tienes clases de las que heredas se aplicarían a LP y BlueRay? Si es así, extiende las nuevas clases. Si no es así, ¿deberías modificarla o ampliar una clase abstracta diferente?
- ¿Qué pasa con las ruedas y los frisbees? También son discos giratorios, pero no se utilizan para el almacenamiento de datos. ¿Cómo entrarían en este modelo de clases?