Principio del formulario

Exame contornos de desenvolvemento Dual - decembro 2023

**masterzono32@gmail.com** [Cambiar de cuenta](https://accounts.google.com/AccountChooser?continue=https://docs.google.com/forms/d/e/1FAIpQLSfCMxjciZv1_x3cqF7D6wPW9-qiXSZQlYSdDY6Umt84eFnqVQ/formResponse&service=wise)

No compartido

Borrador guardado

Cuestiones

El cuestionario consta de 92 preguntas.  
  
- Cada pregunta contestada y fallada resta -0.3  
- Las cuestiones que no se contestan no restan.  
¡Ánimos!

Para iniciar un proyecto con Maven, ejecutas:

mvn site

mvn install

mvn package

mvn archetype:generate

Borrar selección

Para listar las ramas locales que existen en un proyecto, ejecutas:

Cualquiera de las propuestas.

git checkout

git branch

git log

Borrar selección

 La característica de un lenguaje de programación que permite manejar valores de diferentes tipos de datos a través de una interfaz unificada es:

Acoplamiento

Abstracción

Polimorfismo

Herencia

Borrar selección

El principio O de SOLID hace referencia:

A ninguna de las propuestas.

A que los programas deben estar abiertos a la extensión y cerrados a la modificación.

A que la potencia de un programa reside en las estructuras de datos que utiliza.

A que hay que mantener el código sin complejidades excesivas.

Borrar selección

El principio SRP es sinónimo de:

Alto acoplamiento

Abierto a la extensión

Alta cohesión

Todas las propuestas

Borrar selección

Para ignorar el directorio target independientemente de dónde se encuentre situado en el sistema de archivos, incluyes en el fichero gitignore la línea:

Ninguna de las propuestas

!target

foo/\*\*/target

\*\*/target/

Borrar selección

¿Si utilizas Maven, cuál de estos directorios deberías dejar sin seguimiento en Git?

src/main/test

src/main/java

target

Todos los propuestos.

Borrar selección

Maven es:

Una extensión de VSCode

Todas las propuestas.

Una extensión de Eclipse

Un software de administración de proyectos.

Borrar selección

¿Por qué conviene declarar como final los atributos de un objeto que hagan referencia a otros objetos?

Para reducir la mutabilidad del objeto.

Todas las propuestas.

Es una cuestión importante a tener en cuenta en la refactorización de una clase.

Para evitar que los objetos referenciados sean substituidos por otros una vez creado nuestro objeto.

Borrar selección

Un ADT (abstract data type) se define en términos de:

Una representación de los datos.

Un conjunto de operaciones sobre los datos.

Todas las propuestas.

Un conjunto de valores.

Borrar selección

Si quieres añadir al *staging area* todos los archivos de un repositorio git, ejecutas el comando:

git push

git init

git add .

git commit

Borrar selección

Con independencia del IDE o gestor de proyecto Java que utilices, ¿cuál de estos ficheros deberías dejar sin seguimiento en git?

.class

Todas las propuestas

.war

.jar

Borrar selección

¿De qué se habla en una reunión de pie o daily?

Cuáles son tus obstáculos?

Qué completaste ayer?

De qué harás un commit hoy?

De todas las propuestas.

Borrar selección

¿Cuál de los siguientes nombres usarías para nombrar a una rutina que calcula la media de puntos de los equipos de una liga en una determinada jornada?

Ninguno de los propuestos

imprimirMediaPuntos(argumentos)

imprimirMediaJornada(argumentos)

imprimirMedia(argumentos)

Borrar selección

¿Qué nombre recibe el acto de ocultar los detalles funcionales de una clase a los objetos que la usarán posteriormente?

Abstracción

Encapsulación

Polimorfismo

Herencia

Borrar selección

En TDD, ¿qué tipo de código se escribe en la fase roja?

Código que pase el test, aunque no sea del todo eficiente y óptimo.

Código de producción.

Código refactorizado.

Ninguna de las propuestas.

Borrar selección

¿Dónde has de situar el fichero .gitignore?

En el directorio src

Dentro del directorio .git

Es un fichero oculto, así que da igual dónde lo situemos.

En el directorio raíz del proyecto que hayas puesto en seguimiento

Borrar selección

Para saber desde consola qué ficheros se ignoran en el seguimiento git y por qué motivo, ejecutas el comando:

git status -s

Con ninguno de los propuestos

git status --ignored

git status

Borrar selección

¿Cuáles de las siguientes son características de la TDD?

Entender qué hace el código

Mantener el código bajo control

Utiliza técnicas de testing

Todas las propuestas

Borrar selección

¿Cuáles de las siguientes son refactorizaciones admisibles?

Todas las propuestas

Usar un break en vez de una variable de control en un loop.

Reemplazar condicionales con polimorfismo.

Utilizar un return en cuanto sabes la respuesta, en vez de asignar el valor a una variable y devolverla más tarde.

Borrar selección

La repetición de casos test previamente ejecutados con el propósito de encontrar defectos en el software que previamente había pasado los mismos casos test, se llama:

Test de estrés.

Test de rendimiemto.

Test de regresión.

Test de configuración.

Borrar selección

¿Cuándo finaliza un ciclo en la TDD?

Escribiendo código bien diseñado y bien escrito (eficiente y óptimo).

Escribiendo la verificación que ha de pasar el código.

Escribiendo código que pase el test, aunque no sea del todo eficiente y óptimo.

Escribiendo código de producción.

Borrar selección

¿Qué anotación indica que un método ha de ejecutarse antes de cada caso test?

@BeforeTest

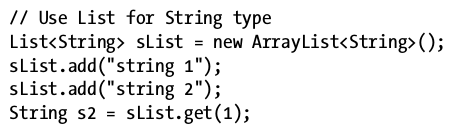
@Test

@Before

@BeforeClass

Borrar selección

¿Qué tipo de polimorfismo se utiliza en este código?



De inclusión (por herencia)

Paramétrico (por genéricos).

Overloading (por sobrecarga)

Coercitivo (por conversión de tipos).

Borrar selección

En TDD, en la fase verde:

Ninguna de las propuestas.

Se escribe código que añada nuevas funcionalidades.

Se mejora el código ya escrito.

Se escribe código que adelante futuras necesidades.

Borrar selección

En una clase con casos test:

El orden de ejecución de los test es aleatorio.

Si utilizamos una estructura de datos para todos los casos test en el método anotado como @BeforeClass, ha de ser declarada estática.

Todas las propuestas son correctas.

Los test han de ser independientes.

Borrar selección

Si una rutina posee más de una responsabilidad, se dice que tiene:

Alta cohesión

Bajo acoplamiento

Baja cohesión

Alto acoplamiento

Borrar selección

¿Qué es debugging / depurar?

Poner de manifiesto los fallos.

Poner de manifiesto los defectos de un programa.

Ninguna de las propuestas es correcta.

Identificar la cadena causa - efecto de un fallo.

Borrar selección

Dada la línea de código Java propuesta, ¿qué puedes deducir?

Impresora.checkCartucho( )

Que Impresora no tiene un constructor.

Que checkCartucho( ) es un método estático.

Que Impresora tiene un constructor por defecto.

Que dentro de checkCartucho( ) se utiliza la referencia this.

Borrar selección

La deuda técnica:

Todas las propuestas son correctas.

Es la acumulación continua de atajos, hacks, código duplicado y otras desgracias que cometemos pretendiendo agilizar la entrega el proyecto.

Se paga refactorizando el código.

Puede tomar la forma de código spaghetti, excesiva complejidad, duplicaciones, etc.

Borrar selección

El concepto de inyección de dependencias, ¿con qué principio SOLID se confunde tradicionalmente y de manera errónea?

ISP

LSP

Con ninguno de los propuestos

DI

Borrar selección

¿Por qué deberíamos usar convenciones para nombrar las variables de nuestro código?

Para garantizar que todos/as escriban código utilizando el mismo idioma.

Todas las propuestas.

Para utilizar en el proyecto siempre las mismas abreviaturas que resultan difíciles de entender.

Para tomar una única decisión global y no múltiples locales.

Borrar selección

¿Cuál de las siguientes ramas no forma parte de la propuesta gitflow?

release

test

develop

hotfixes

Borrar selección

¿Cuáles de las siguientes son características de la TDD?

Que el programador mantenga el código escrito bajo control.

Todas las propuestas

La integración continua de código en el producto.

Escribir el máximo de código posible.

Borrar selección

¿Qué nombre recibe el componente a partir de la cual se instancia un objeto?

Interfaz

Entidad

Instancia

Clase

Borrar selección

¿La depuración se incluye en qué fase del ciclo de vida del software?

En todas las propuestas.

Pruebas

Mantenimiento

Implementación

Borrar selección

¿Con qué otros dos conceptos va de la mano la cohesión?

Con ambos

Abstracción

Encapsulamiento

Con ninguno de los propuestos.

Borrar selección

¿Qué código ejecuta la máquina virtual de Java?

Bytecode

Fuente

Ensamblador

Objeto

Borrar selección

Si quieres consultar la historia de commits de un repositorio git en tu máquina, ejecutas:

git diff --staged

git status -s

git diff

git log

Borrar selección

¿Qué crees que significa la idea *convention over configuration* o convención sobre la configuración, que está en el corazón de Maven?

Que a los desarrolladores no se les exige crear el proceso de construcción ya que Maven provee de comportamiento por defecto para cada tipo de proyecto.

Que Maven genera una estructura de directorios estándar para un proyecto Java y una ciclo de vida por defecto.

Es un paradigma que intenta disminuir el número de decisiones que debe realizar el programador/a sin perder flexibilidad.

Todas las propuestas

Borrar selección

¿Para añadir una nueva librería al proyecto mediante Maven, en qué sección del POM.xml  has de añadir una nueva entrada?

<dependencies>

<configuration>

<plugins>

<build>

Borrar selección

¿Cuál o cuáles de las siguientes NO constituyen un nombre de rutina adecuado?

Todas NO constituyen un nombre adecuado.

printTuple( ... )

delIndexList( ... )

cambiarClaveValor( ... )

Borrar selección

¿Cuáles de los siguientes son principios clave en el diseño de un programa?

Bajo acoplamiento entre rutinas

Una rutina debería ser responsable de una única función.

Una rutina no debería conocer detalles internos de otras rutinas.

Todos los propuestos.

Borrar selección

Para realizar una instantánea del estado de tu proyecto y guardarla en el historial de versiones, en git utilizas el comando:

git add .

git commit -m "comentario"

git diff

git push

Borrar selección

¿Qué es una infección?

Un valor no válido de una de las variables del programa.

Una sentencia del programa que contiene un defecto.

La ejecución de un fallo.

Un bug.

Borrar selección

Cuando trabajas con JUnit para escribir casos test, estás realizando:

Pruebas de caja blanca.

Beta testing.

Pruebas de caja negra.

Test de usabilidad.

Borrar selección

En las reuniones de pie o dailies (en Scrum), se persigue:

Que todo el equipo observe y comprenda el estado del proyecto.

Empezar el día "bien".

Crear sentimiento de equipo.

Todas las propuestas.

Borrar selección

Una sentencia break dentro de un bucle:

Puede escribirse tanto en los bucles while como for

Permite que el bucle sea tratado por el programador como una caja negra.

Ninguna de las propuestas.

Se ejecuta tantas veces como iteraciones realice el bucle.

Borrar selección

Si quieres obtener una copia de un repositorio remoto en un directorio local de tu máquina, ejecutas:

git init <URL del proyecto>

git clone <URL del proyecto>

git pull

git remote

Borrar selección

¿Qué es testing / testear?

Identificar la cadena causa - efecto de un fallo.

Poner de manifiesto los defectos de un programa.

Romper la cadena causa - efecto de un fallo.

Poner de manifiesto los fallos.

Borrar selección

¿Desde qué punto de vista se evalúa si una historia de usuario/a es valiosa?

Desde el punto de vista del scrum master.

Desde el punto de vista del equipo de desarrollo.

Desde el punto de vista del comprador del software.

Según su dificultad técnica.

Borrar selección

¿Cuáles de estas acciones pueden llevarse a cabo con Maven?

Todas las propuestas.

Testear el proyecto.

Crear un fichero de distribución jar.

Compilar el proyecto.

Borrar selección

¿Qué contiene el fichero POM.xml?

Contiene detalles de configuración usados por Maven para construir el proyecto.

Todas las propuestas

Contiene las dependencias del proyecto.

Contiene valores por defecto para la mayoría de los proyectos.

Borrar selección

Indica qué deben cumplir los casos test:

Han de ser automáticos.

Han de cubrir la mayor parte del código.

Todos los propuestos.

Han de cubrir todos los posibles casos.

Borrar selección

¿Cuál de los principios SOLID asociarías con la sentencia "Un módulo debe ser responsable (de algo) ante un solo actor".

LSP

Ninguno de los propuestos.

OCP

SRP

Borrar selección

Si has añadido en los ficheros CSS de tu proyecto las *media query*para conseguir que tu sitio sea *responsive,* ¿cuál sería el mensaje del *commit* para respetar la convención *conventional commits?*

"style(media.css): Incluir media query dispositivos moviles"

"feat(media.css): Incluir media query dispositivos moviles"

"fix(media.css): Incluir media query dispositivos moviles"

"refactor(media.css): Incluir media query dispositivos moviles "

Borrar selección

Para eliminar la rama spam en git ejecutas el comando:

Ninguna de las propuestas.

git branch -d spam

git merge spam

git checkout -b spam

Borrar selección

¿Qué linea has de añadir al fichero .gitignore para evitar el seguimiento del directorio .vscode ?

.vscode/

# .vscode

Se trata de un directorio oculto y, por tanto, por defecto git no hace seguimiento.

.vscode \*

Borrar selección

Tienes que programar la actualización de la calidad de los items de un stock o inventario. Si tu código depende de sus tipos específicos en vez del tipo más general, ¿qué principio SOLID no se ha respetado?

ISP

SRP

OCP

DI

Borrar selección

Cuando averiguas la causa de un error y modificas el código, has hecho:

Debugging

Testing

Una prueba de caja negra.

Todas las propuestas.

Borrar selección

¿Cuál de estos es un comando adecuado para ejecutar una aplicación Java desde su jar?

java -jar target/refactorex-1.0-SNAPSHOT.jar

jar target/refactorex-1.0-SNAPSHOT.jar

mvn run

mvn compile

Borrar selección

Cuando en JUnit escribes los test que forman parte de la TDD para codificar una determinada clase que no involucra funcionalidades de otras clases, estás haciendo:

Unit testing.

System testing.

Integration testing.

Component testing.

Borrar selección

¿Cuáles de los siguientes mensajes de *commit* respetan la convención *conventional commit*?

Ninguno de los propuestos

"testing(test\_a\_nulo): Añadir casos test coeficiente a nulo"

"refactor(media.css): Incluir media query dispositivos 600px 700px 800px 1200px"

"fix(raiz\_cuadrada): fixing bug coeficiente a nulo"

Borrar selección

Si una clase realiza varias operaciones poco relacionadas entre si, se dice que tiene:

Bajo acoplamiento

Alto acoplamiento

Baja cohesión

Alta cohesión

Borrar selección

El directorio .git

Contiene los ficheros de configuración de git para ese repositorio.

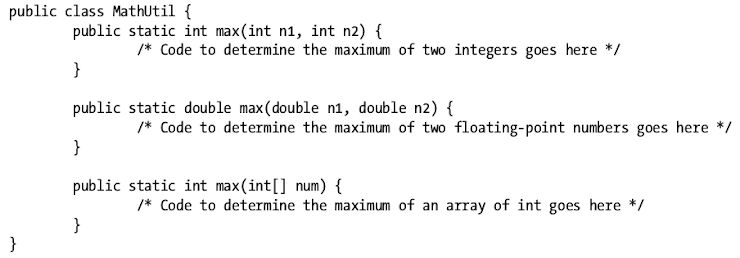
Contiene la estructura del repositorio git

Es creado por el comando git init

Todas son correctas.

Borrar selección

¿Qué tipos de polimorfismo exhibe este código?



Paramétrico (por genéricos).

Overloading (por sobrecarga)

Coercitivo (por conversión de tipos).

De inclusión (por herencia)

Borrar selección

Para crear la rama spam en git ejecutas el comando:

Ninguno de los propuestos

git branch -d spam

git branch spam

git checkout spam

Borrar selección

Es deseable una rutina con:

Aquella que no hace suposiciones sobre los usuarios/as de la rutina.

Todas las propuestas

Bajo acoplamiento

Alta cohesión

Borrar selección

¿Qué objetivos persigue la refactorización del código?

Todas las propuestas.

Que los futuros cambios salgan baratos

No cambiar su comportamiento

Hacerlo fácil de entender

Borrar selección

¿Cuáles de los siguientes son principios clave en el diseño de un programa mediante programación orientada a objetos?

Bajo acoplamiento entre clases.

Todos los propuestos.

Una clase no debería conocer detalles internos de otras clases.

Una clase debería presentar alta cohesión.

Borrar selección

¿Qué significan las siglas TDD?

Task Driven Development

Time Driven Development

Total Doomed Development

Test Driven Development

Borrar selección

¿Cuáles de las siguientes no forman parte del JRE de Java?

Java Virtual Machine (JVM)

Java API

Class loader

JDK

Borrar selección

Si quieres listar las ramas remotas y las locales, ejecutas:

git branch -a

git branch -r

git branch -v

Con todas las propuestas

Borrar selección

¿Cuándo has de usar un bucle for?

Si conoces el número exacto de veces que se ejecutará el bucle.

Cuando quieres recorrer una colección de objetos.

Cuando el bucle no requiere sentencias internas de control.

Todas las propuestas.

Borrar selección

Para crear una nueva rama en git y saltar a ella, en un solo paso, puedes utilizar el comando:

git branch rama\_nueva

Con ninguno de los propuestos.

git checkout rama\_nueva

git checkout -b rama\_nueva

Borrar selección

¿Qué encontramos en un archivo jar de Java?

Todos los propuestos

Bytecode

El archivo Manifest.

La clase principal de la aplicación.

Borrar selección

¿Cuándo comienza el ciclo de la TDD?

Escribiendo código de producción.

Escribiendo código que pase el test, aunque no sea del todo eficiente y óptimo.

Escribiendo la verificación que ha de pasar el código.

Escribiendo código bien diseñado y bien escrito (eficiente y óptimo).

Borrar selección

¿Cuáles de las siguientes son motivos para refactorizar?

Una clase usa sólo una pequeña parte de las rutinas de la clase de la que hereda.

Todos los propuestos.

Se han usado variables globales.

Se utilizan comentarios para explicar el código.

Borrar selección

Si quieres añadir y hacer un commit de los cambios del directorio git al mismo tiempo, ejecutas el comando:

git commit -m 'terminado footer [ticket 112]'

Cualquiera de las propuestas.

git commit -a 'terminado footer [ticket 112]'

git commit -a -m 'terminado footer [ticket 112]'

Borrar selección

Una clase es:

Una herramienta para simular y manipular entidades del mundo real

Todas las propuestas.

Una colección de datos y operaciones que trabajan sobre esos datos.

Un tipo abstracto de dato.

Borrar selección

¿Cuáles de las siguientes no forman parte de la TDD?

Todas forman parte

Depurar

Testear

Refactorizar

Borrar selección

¿Cuál de los siguientes es un esquema de nombres adecuado para nombrar variables y rutinas, es decir, cuál de las siguientes expresiones utilizarías para nombrar variables y rutinas?

Todos los propuestos.

[cliente.id](https://www.google.com/url?q=http://cliente.id&sa=D&source=editors&ust=1710506446876931&usg=AOvVaw1SAXqjt8kN5coW7MTeMgdn)

clienteId

cliente\_id

Borrar selección

¿Cuándo deberíamos definir una convención para los nombres de las variables?

Todas las propuestas.

Cuando existen múltiples términos inusuales que aparecen muchas veces en el código, y quieres abreviarlos.

Cuando el código va a ser revisado por varias personas.

Cuando trabajan múltiples programadores/as en el proyecto.

Borrar selección

Una rutina que devuelve un valor recibe el nombre de:

Procedimiento

Método

Ninguno de los propuestos

Función

Borrar selección

Cuando ejecutas: mvn clean

Se elimina el directorio bin

Se elimina el directorio test

Todas las propuestas son correctas.

Se elimina el directorio target

Borrar selección

Indica qué sentencia es correcta:

El bytecode de una clase Java compilada no depende de la máquina donde se compila.

Todas son correctas.

El "lema" de Java es "Write once, run anywhere".

El bytecode se ejecuta en una máquina virtual de Java que sí depende de la arquitectura donde se instala.

Borrar selección

¿Cuál de las siguientes características debería tener el nombre de una variable?

Ninguna de las propuestas

Tener menos de 10 caracteres

Utilizar siempre el guión bajo para separar palabras o conceptos.

Ser un mnemónico (técnica de memorización basada en la asociación mental de la información a memorizar con datos que ya sean parte de nuestra memoria).

Borrar selección

Las reuniones de pie o dailies se celebran:

Todas las propuestas son correctas.

Por ejemplo, al empezar el día.

Delante de un panel de actividades.

Cuando el grupo decida, preferiblemente en el mismo lugar y a la misma hora.

Borrar selección

Si una rutina realiza una única operación, se dice que tiene:

Alto acoplamiento

Bajo acoplamiento

Alta cohesión

Baja cohesión

Borrar selección

La sigla "S" en SOLID significa:

Single Responsability

Una rutina ha de tener un único motivo para cambiar

Alta cohesión

Todas las propuestas son correctas

Borrar selección

¿Qué comando se ejecuta para poner en seguimiento con git el directorio actual?

git status

git

git init

git pull

Borrar selección

La instrucción *pass* de Java en un *else*:

Puede acompañar a comentarios en el código para explicar por qué la clausula else no es necesaria.

Es una técnica de control de las estructuras condicionales.

No existe la palabra reservada pass en Java

Muestra que el programador ha considerado el caso else.

Borrar selección

Atrás

Enviar

Borrar formulario

Nunca envíes contraseñas a través de Formularios de Google.

Final del formulario

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](https://docs.google.com/forms/u/0/d/e/1FAIpQLSfCMxjciZv1_x3cqF7D6wPW9-qiXSZQlYSdDY6Umt84eFnqVQ/reportabuse?source=https://docs.google.com/forms/d/e/1FAIpQLSfCMxjciZv1_x3cqF7D6wPW9-qiXSZQlYSdDY6Umt84eFnqVQ/viewform) - [Términos del Servicio](https://policies.google.com/terms) - [Política de Privacidad](https://policies.google.com/privacy)

[Formularios](https://www.google.com/forms/about/?utm_source=product&utm_medium=forms_logo&utm_campaign=forms)