



INVESTIGACIÓN

Sharon Belinda Toral Rodríguez



3 DE MARZO DE 2023

ACADEMIA JAVA
Xideral / Accenture

Investiga los comandos de Git:

Pull request

Este comando permite hacer una solicitud de `merge` a los colaboradores de un proyecto en el que estén trabajando, por medio de la solicitud los implicados podrán revisar los cambios implementados en la rama a fusionar, informarse y/o dar sugerencias, una vez que todo el equipo aprueba el `pull request` se puede realizar el `merge`.

Fork

Es una herramienta de GitHub que te permite crear una copia de un repositorio de otro usuario en el repositorio personal, abriendo la posibilidad de adquirir y editar el código de otros.

Rebase

Permite integrar los cambios de una rama a otra, re-escribiendo el código del que parte una rama.

Stash

Permite guardar cambios de forma local y temporal al mismo tiempo que deja el directorio de trabajo limpio (sin cambios), se usa comúnmente cuando se ejecutan cambios en la rama equivocada pero no se quieren perder de estos. Posteriormente, ya que se encuentra en la rama adecuada se usa `git stash pop`, este comando trae los últimos cambios guardados en el `stash`.

Git clean

Borra los archivos sobre los que no se ha hecho un `git add`, es decir trabaja sobre archivos que aún no han sido agregados al área de almacenamiento temporal (antes de hacer de un `add`, staging area), una vez que se ejecuta no se puede deshacer.

Opciones:

- `-n`, muestra los archivos a borrar antes de borrarlos
- `-d`, borra directorios no agregados
- `-f`, borra y muestra los archivos que borró

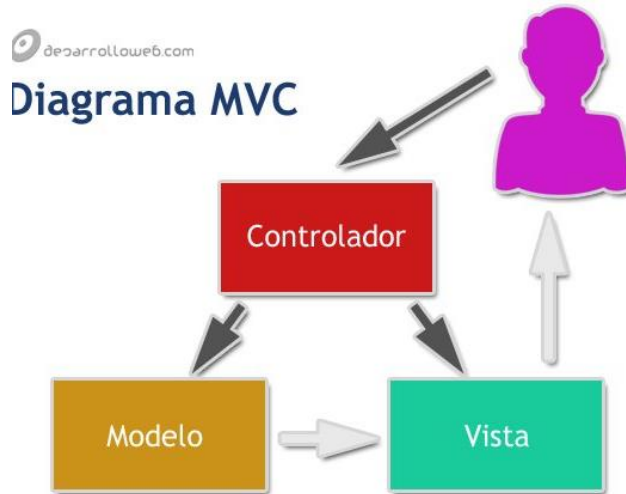
Cherry pick

Permite tomar los cambios relacionados a un `commit` en específico, de una rama de desarrollo, cuyo id se puede conocer, e integrarlo a la rama principal, sin integrar todos los elementos de la rama.

Explicar y diagramar MVC

MVC (modelo-vista-controlador) es una estructura de desarrollo de software, comúnmente usada en sistemas que requieren de una interfaz de usuario, esta arquitectura de software divide el código, separando las responsabilidades en tres capas:

- **Modelo**, este bloque está a cargo de los datos y los mecanismos que implican su uso
- **Vistas**, su función gira en torno a la interacción con el usuario, es decir, el uso de interfaces, este bloque no puede acceder por sí solo a los datos
- **Controladores**, es la conexión entre el bloque de vistas y modelo



Diferencias entre aplicaciones monolíticas y microservicios

Ambas son estructuras de desarrollo de software, actualmente hay una tendencia de pasar de una estructura monolítica a una de microservicios.

Monolítica:

- Modelo tradicional
- un programa de software que se compila como una unidad unificada
- es autónoma e independiente de otras aplicaciones
- Red informática unificada
- Difícil de mantener
- Simple de usar
- Poca escalabilidad y flexibilidad

Microservicios:

- Separar en partes el software dependiendo de su objetivo y/o funcionamiento
- separan las tareas en procesos más pequeños e independientes entre sí
- Bueno para software en crecimiento o de gran escala
- Actualizaciones frecuentes
- Alta escalabilidad y flexibilidad
- Complejo de usar

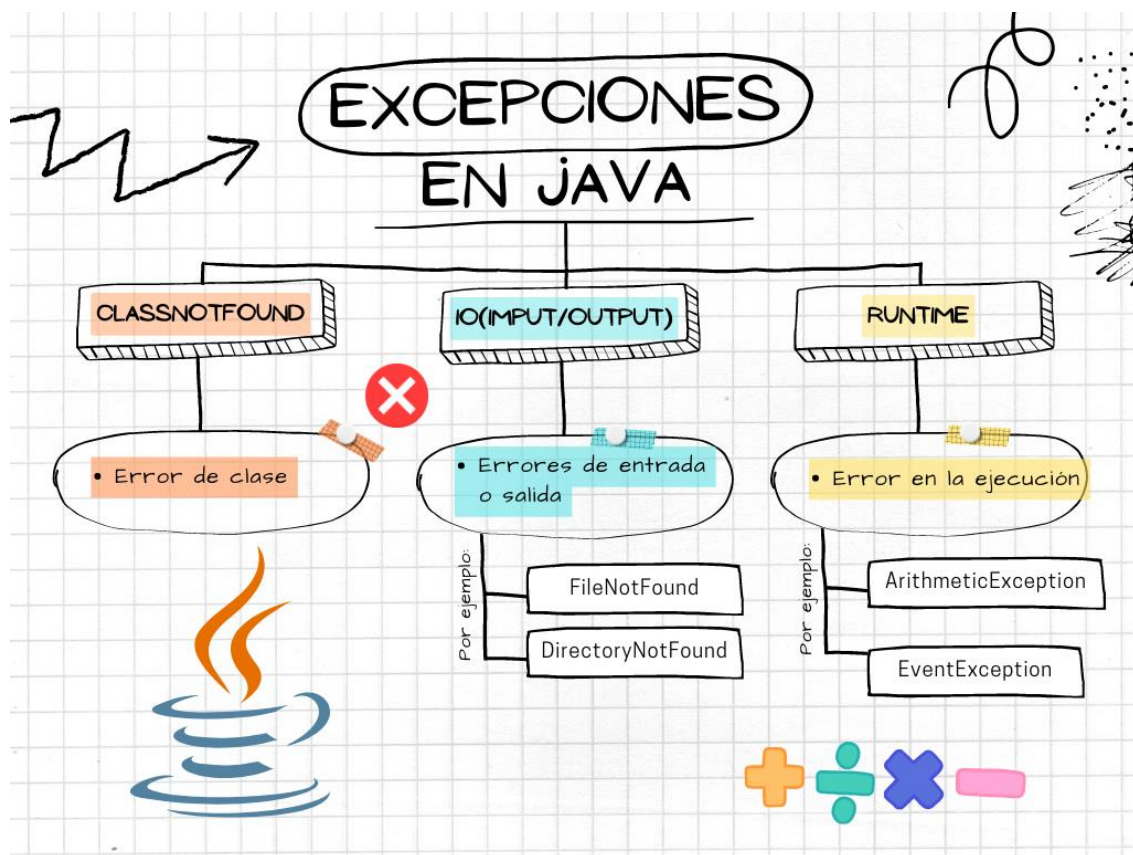
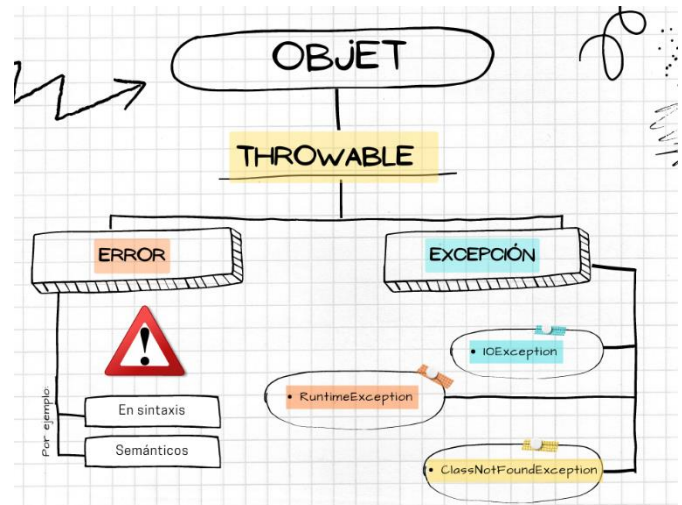
Explica y diagrama los 3 tipos de excepciones en Java

Las excepciones son una herramienta por la cual podemos identificar errores o fallas en nuestros programas y decidir cómo tratarlos. En java estos "errores" como objetos dentro de una jerarquía de clases

Throwable es un objeto que puede ser "lanzado" agrupa excepciones y errores.

Los **errores** son aquellos fallos sobre los que el programador no tiene incidencia y no permiten la ejecución del problema.

Las **excepciones** son fallos que se buscan solucionar mediante el lanzamiento de alertas al usuario, se clasifican en tres grandes grupos: error de entrada o salida, de tiempo en ejecución y de clase.



Explica el try with resource y el multicatch

Multi-catch: `catch(err | err | err nombre)`

Dentro del tratamiento de excepciones se usa se utiliza la sentencia catch para "tomar" el error esperado, asignarle un nombre y un tratamiento, dentro de nuestro código es posible que surjan más de una excepción, y si la forma en que los solucionamos es similar podemos usar una sentencia multi catch, que nos ayuda a reducir el código repetido y regresa la respuesta a una sola excepción.

Try-with-resources: `try(new resource)`

En esta sentencia se declaran uno o más recursos, es decir objetos que deber ser "cerrados" al finalizar el programa. Los objetos llamados con esta sentencia son aquellos que pueden implementar el método AutoClosable.

Referencias:

- Gandhi, R. (2022). *Head First Git*. Van Duuren Media. Pp 218-219, 254, 445-447, 449-450
- Atlassian. (s. f.). *Git Clean | Atlassian Git Tutorial*.
<https://www.atlassian.com/git/tutorials/undoing-changes/git-clean>
- The try-with-resources Statement (The Java™ Tutorials > Essential Java Classes > Exceptions). (s. f.).
<https://docs.oracle.com/javase/tutorial/essential/exceptions/tryResourceClose.html>
- Catching Multiple Exception Types and Rethrowing Exceptions with Improved Type Checking. (s. f.).
<https://docs.oracle.com/javase/7/docs/technotes/guides/language/catch-multiple.html>
- Qué es MVC. (2020, 28 julio). Desarrollo Web.
<https://desarrolloweb.com/articulos/que-es-mvc.html>
- MVC - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web | MDN. (2022, 5 diciembre).
<https://developer.mozilla.org/es/docs/Glossary/MVC>
- Atlassian. (s. f.-a). *Comparación entre la arquitectura monolítica y la arquitectura de microservicios*. <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>
- *Uso de excepciones en Java*. (2021, 18 noviembre). IfgeekthenNTTdata.
<https://ifgeekthen.nttdata.com/es/uso-de-excepciones-en-java>
- Martin, E. (2014, 5 septiembre). *Tipos de excepciones en Java*. Solvetic.
<https://www.solvetic.com/tutoriales/article/1136-tipos-de-excepciones-en-java/>