

### Tercer examen Xideral

#### Primera pregunta:

Escribe un escenario donde crearas dos ramas, la principal master y otra rama que sea clon de la principal, ambas tendrán el mismo archivo, se modificara él mismo de diferente manera en ambas ramas para que al realizar un merge se produzca un conflicto. Describe los pasos para la resolución del conflicto.

#### Respuesta:

Los comandos utilizados se encuentran en el archivo: Comandos usados para la primer pregunta.txt

Descripción de los pasos previos:

Se creo un repositorio remoto en Git hub con el nombre TercerExamenXideral.

Ubicándonos en una carpeta en Documents con el nombre TercerExamenXideral, se utilizó Git Bash.

Primer parte:

Se utilizo "git init" para iniciar un repositorio remoto en esta ubicación, "git remote add origin más la dirección del repositorio remoto" para enlazar al repositorio remoto, "mkdir Practica Git" creara una carpeta con el nombre Practica Git, entrando a la carpeta se utilizó "touch Apuntes de examen.txt" para crear un archivo txt, "notepad más el nombre del archivo" abre el archivo en bloc de notas, insertaremos unas líneas de texto, guardaremos y cerraremos. "git add ." añadirá el archivo al índice del repositorio, "git commit -m "mensaje" " guardara el archivo en la base del repositorio, "git push origin master" enviara-guardara la información en el repositorio remoto master.

Segunda parte:

"git branch master\_clon" creara una rama con el nombre master\_clon que contiene los mismos archivos que la rama master, "git checkout master" para movernos a la rama master\_clon (también puede utilizarse switch). Se modificó el archivo "Apuntes de examen.txt" agregándole una línea de texto y se guardó.

"git add ." para añadir el archivo al índice, "git commit -m "mensaje"" para añadir el archivo a la base del repositorio, "git push origin master\_clon" " enviara-guardara la información en el repositorio remoto en la rama master\_clon.

Regresamos a la rama master con "git checkout master". Se modificó el archivo "Apuntes de examen.txt", se añadió al índice con "git add .", se realizó un "commit" y se subió el archivo a la rama master con "git push origin master".

Tercer parte:

Utilizando "git merge master\_clon" desde la rama master, para mezclar las ramas, esto causara un conflicto, ya que el archivo "Apuntes de examen.txt" contiene información diferente en ambas ramas. Al ejecutar "git diff" nos muestra donde está el conflicto y las líneas de código que hay en la rama master y en la rama master\_clone.

Cuarta parte:

Nos dirigimos a la carpeta que contiene el archivo y lo abrimos con Notepad para revisar el conflicto:

++<<<<<< HEAD

+Mi primer línea de texto

Javier de Jesús Pérez Santiago

- -Mi segunda línea de texto en la rama master

++-Mi segunda línea de texto en la rama master

++=====

+ -Mi primer línea de texto

+

+ -Mi tercer línea de texto desde mi rama master\_clone

++>>>>>> master\_clon

Para eliminar el conflicto se debe elegir entre la información de la primera rama master(head) o el de la segunda (master\_clon), una vez elegida, se debe borrar la información que no se quiere y eliminar las flechas de separación que nos escribió Git, guardamos el archivo y cerramos. Añadimos el archivo con "git add .", utilizamos "git commit" y ejecutamos "git push origin master" para subir los cambios, con esto se eliminaría el conflicto.

### **Segunda pregunta:**

Realiza una calculadora con funciones lambda y polimorfismo.

### **Tercera pregunta:**

Dibuja un diagrama de servicios Rest Full

### **Cuarta pregunta:**

Diseña una aplicación con Spring Boot utilizando Hibernate y JDBI.

### **Quinta pregunta:**

Explica que es la inyección de dependencias

### **Respuesta:**

En java la dependencia nos indica la relación que existe entre dos clases. Como su nombre indica nos esta diciendo que una clase depende de otra para realizar su funcionamiento.

La inyección de dependencias es un patrón de diseño de programación orientada a objetos cuya finalidad es la de suministrar objetos a una clase mediante otra clase o en el caso de Spring mediante el contenedor de Spring, sin que la propia clase tenga que crearlos.

Cuando un objeto depende de otro no debe depender de la implementación de ese objeto, debe depender de una implementación del mismo, no de una idea específica (un objeto) si no de una abstracción de ese objeto.

Ventajas:

-Se le suministran objetos a una clase, en vez de que una clase sea la que cree dichos objetos

-Nuestras clases no crean los objetos, si no que se los suministran otras clases (por lo principal la main) que implementara la implementación deseada.

-Minimiza la dependencia de nuestras clases.

Javier de Jesús Pérez Santiago

- Mejora los procesos de testing.

- Solo es pasar (inyectar) las dependencias (objetos) cuando sea necesario en lugar de inicializar las dependencias (objetos) dentro de la clase receptora.

Existen 3 tipos de inyección de dependencias:

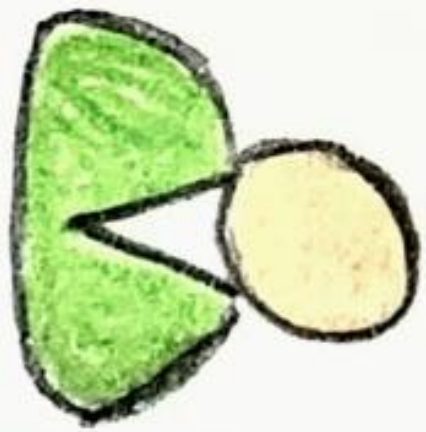
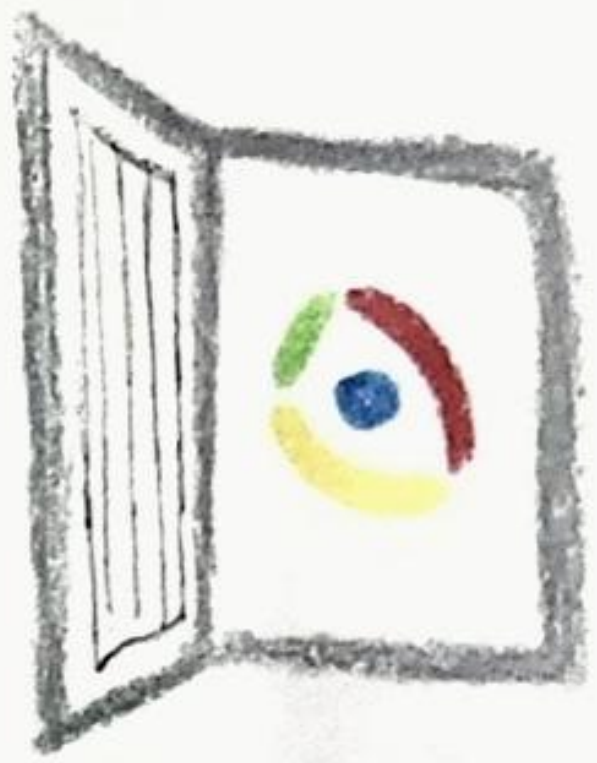
- Por variable o campo

- Por setter

- Por constructor



Peticiones Get



React



ANGULAR

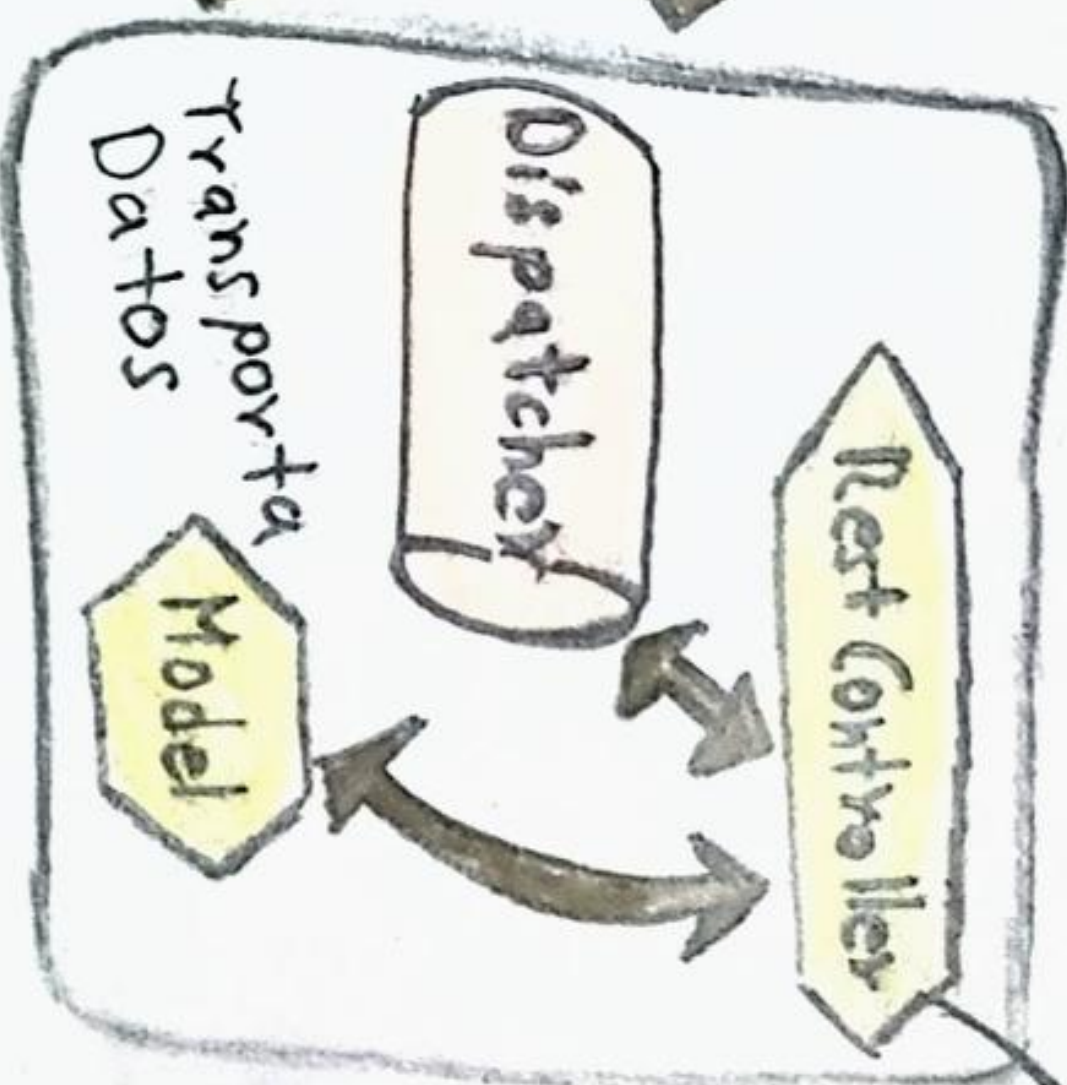


Postman



request  
response  
+txt  
JSON  
Xml, etc.

Http/Https  
get  
post  
put  
delete, etc



Api Rest

Dirige  
Controla  
Asigna

Capa de  
Servicios  
(Java)

Lógica  
de  
Negocio

Transaccional

Capa de  
Persistencia  
(Java)

ORM  
-Hibernate  
-JPA  
-etc

Java  
JDBC  
API

Lenguaje  
Desarrollo  
procedural



SQL

Base de datos relacionales  
MySQL / Oracle / PostgreSQL  
Sql Server Microsoft +