

Erick Antillón

Introducción a la programación y computación 1

Clase 5- Continuando con Git y POO en Java



Agenda

- Calificación practica 1
- Manejo de memoria
- Manejo de Excepciones y Debugging
- Introducción al control de versiones
- Git y Github
- Ejemplo usando git
- Tarea #1
- Asistencia

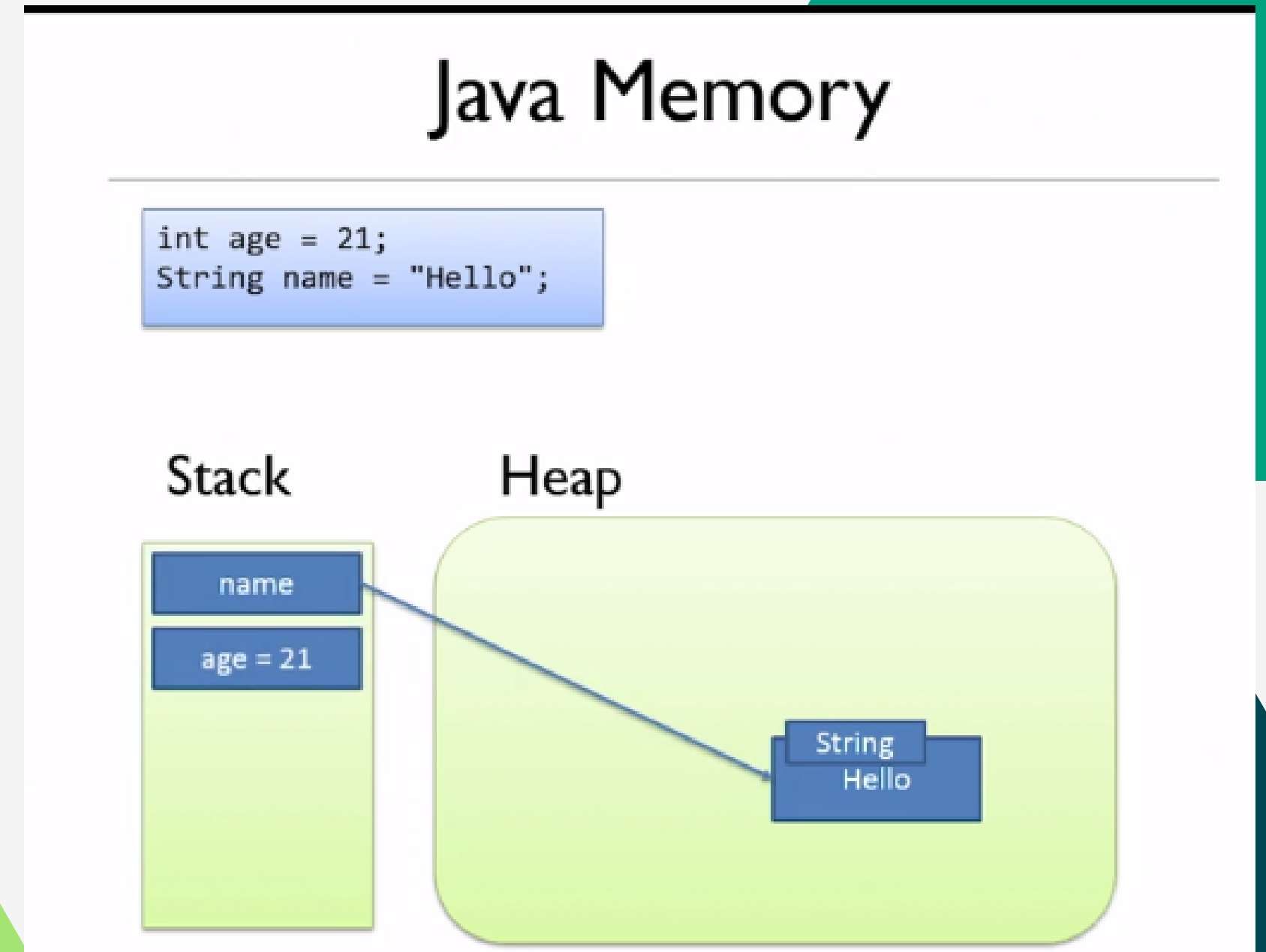
Dudas

Practica #1



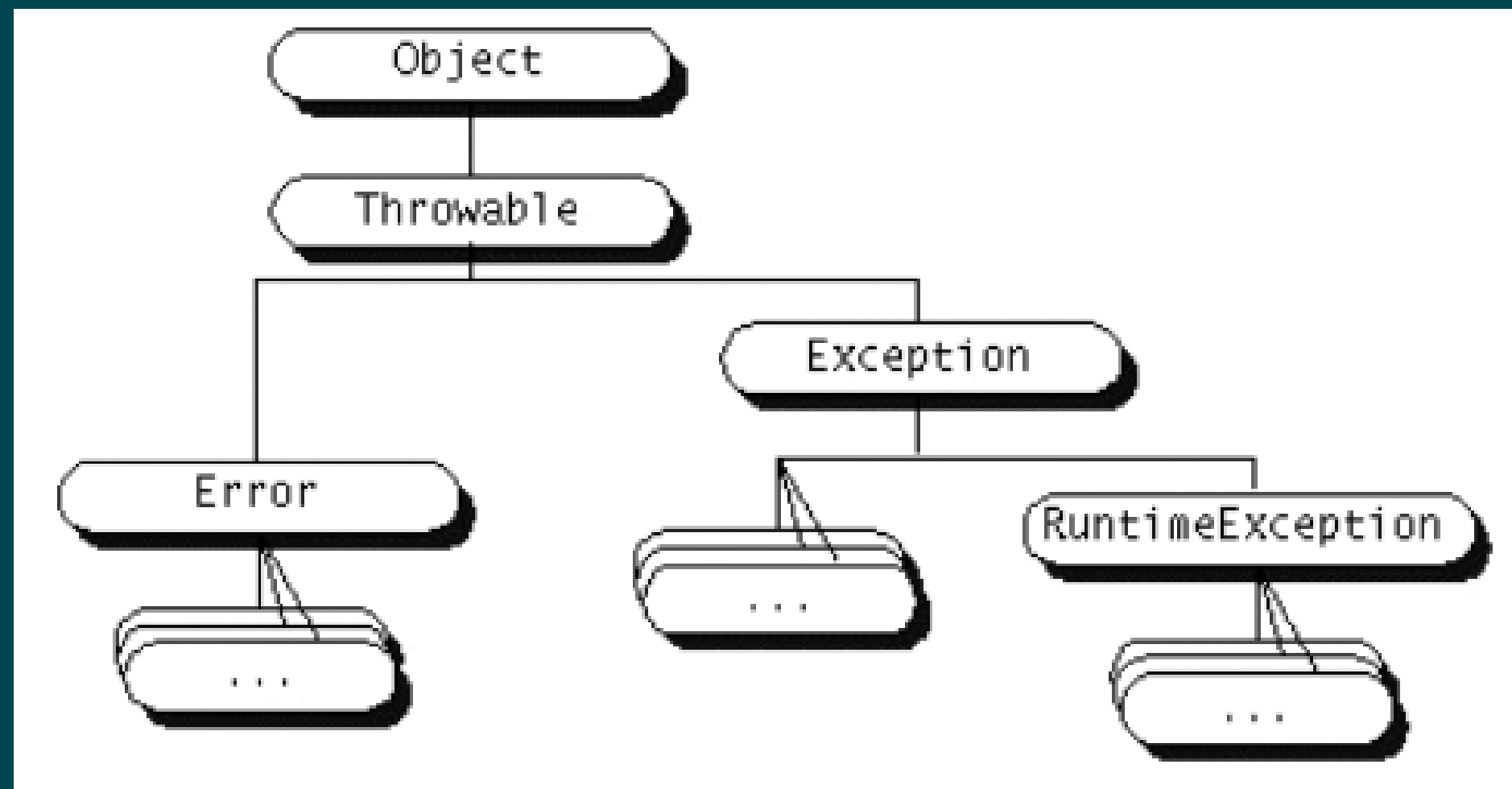
Manejo de memoria en Java

- Java utiliza un modelo de memoria administrada por el "Garbage Collector". Este se encarga de liberar automáticamente la memoria que ya no se utiliza.
- La memoria se divide en dos áreas principales: stack y heap.
- El stack se utiliza para almacenar información de corto plazo y el heap se utiliza para almacenar objetos y datos de largo plazo.



Excepciones

- Java es un lenguaje compilado, por tanto durante el desarrollo pueden darse dos tipos de errores: los de tiempo de compilación y los de tiempo de ejecución.



Excepciones

Errores de compilación

```
public class Main {  
    public static void main(String[] args) {  
        fo ( int i = 0; i < 4; i++ )  
        {  
            // codigo  
        }  
    }  
}
```

Errores en tiempo de ejecución

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Antes del error");  
        int numerador = 10  
        int denominador = 0;  
        int division = numerador/denominador; // Error  
        System.out.println("Ya no llego aqui :(");  
    }  
}
```

Manejo de errores

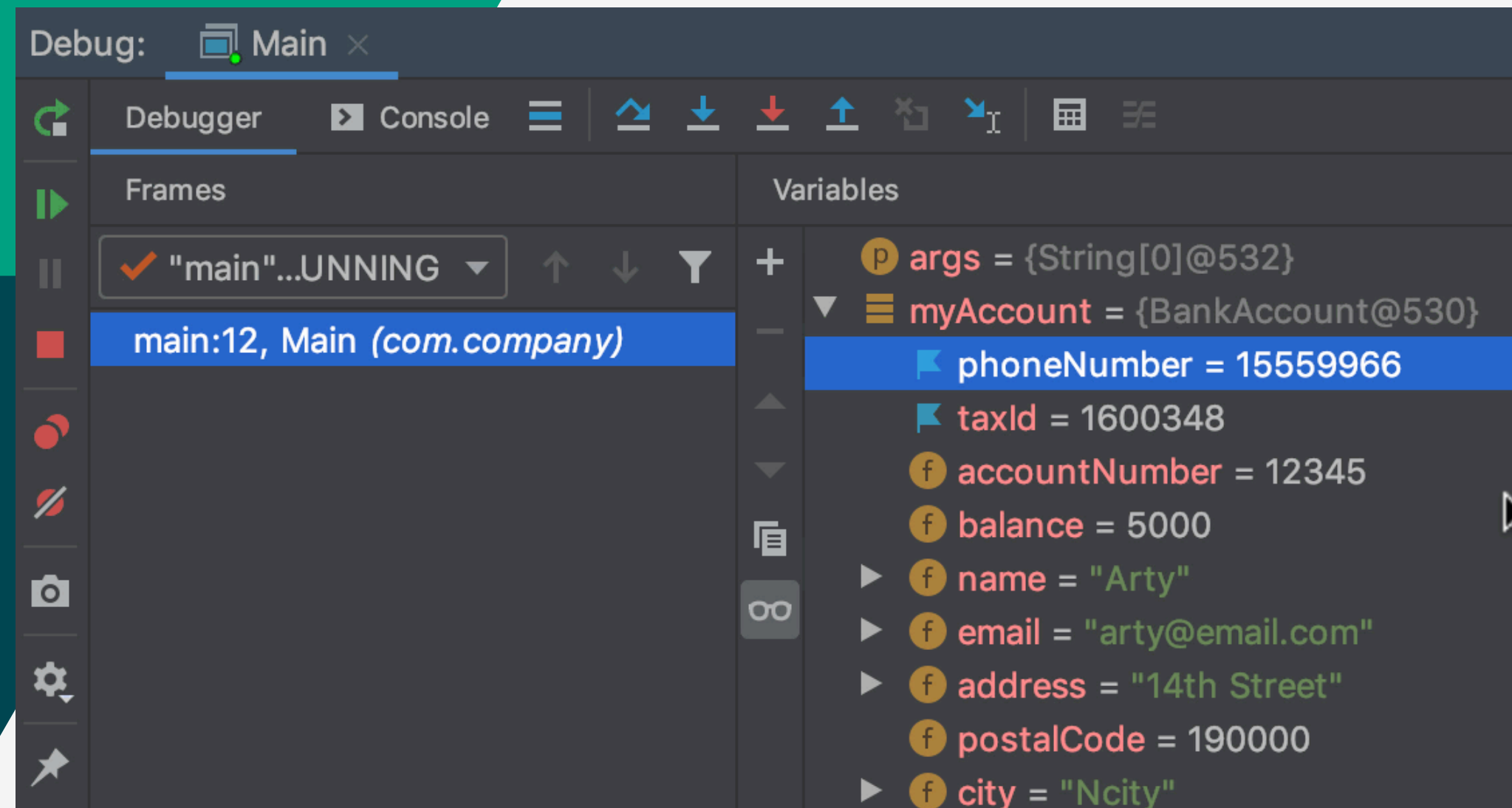
- La estructura básica de control de errores en Java es la siguiente:

```
try {  
    // código que puede generar una excepción  
} catch (ExceptionType e) {  
    // código para manejar la excepción  
} finally {  
    // código opcional que se ejecuta independientemente de  
    // si se genera una excepción  
}
```

Herramienta de Debugg

Los IDE suelen proporcionar una serie de herramientas de depuración para ayudar a los desarrolladores a encontrar y solucionar errores en su código. Algunas de las herramientas de depuración más comunes:

- Punto de interrupción
- Inspección de variables
- Evaluación de expresiones
- Seguimiento de la ejecución
- Registro de excepciones



Control de versiones

Existen herramientas que permiten a los desarrolladores llevar un seguimiento de los cambios realizados en el código de un proyecto a lo largo del tiempo.

Hay muchas herramientas disponibles, pero una de las más populares es Git.

Git es un sistema de control de versiones distribuido que permite almacenar y gestionar diferentes versiones de un proyecto de manera eficiente.



Plataformas de alojamiento de repositorio (git)

- Existen distintas plataformas web para alojar proyectos de software y colaborar en el desarrollo de código.

Github

Lanzada en 2008. Es la plataforma de alojamiento de repositorio basada en git, más popular. Adquirida por Microsoft en 2018.



Gitlab

Lanzada en 2011. Desde el principio quería diferenciarse de Github, por lo que creó un producto para todo el ciclo de vida de DevOps



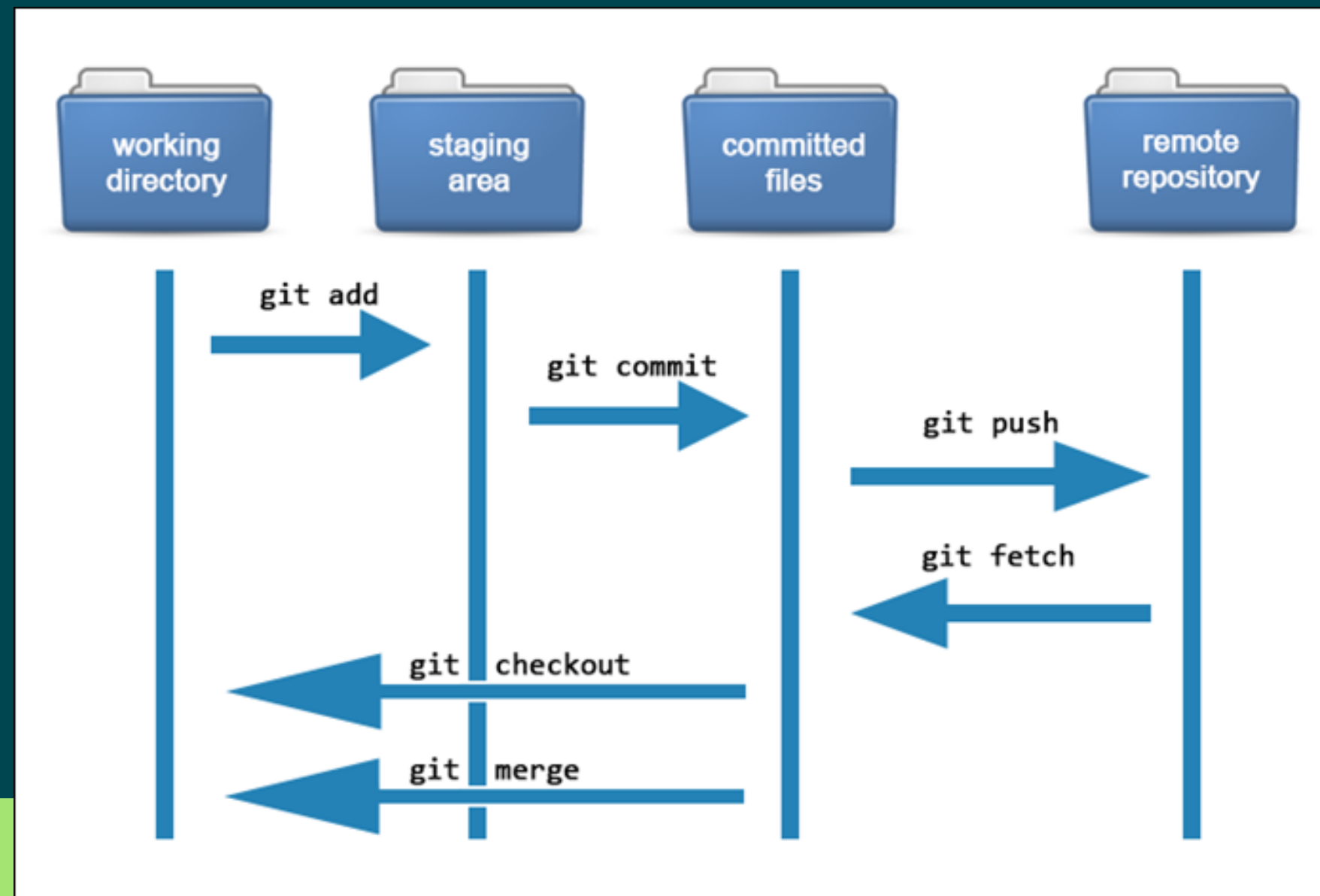
Bitbucket

Diseñado para que lo utilicen equipos que aprovechan Jira de Atlassian.



¿Cómo funciona Git?

Cada proyecto en Git tiene su propio repositorio, que es una colección de archivos y directorios que se controlan con Git.



Comandos más usados

- git clone
- git branch
- git checkout
- git status
- git add
- git commit
- git push
- git revert
- git merge



10 Git Commands

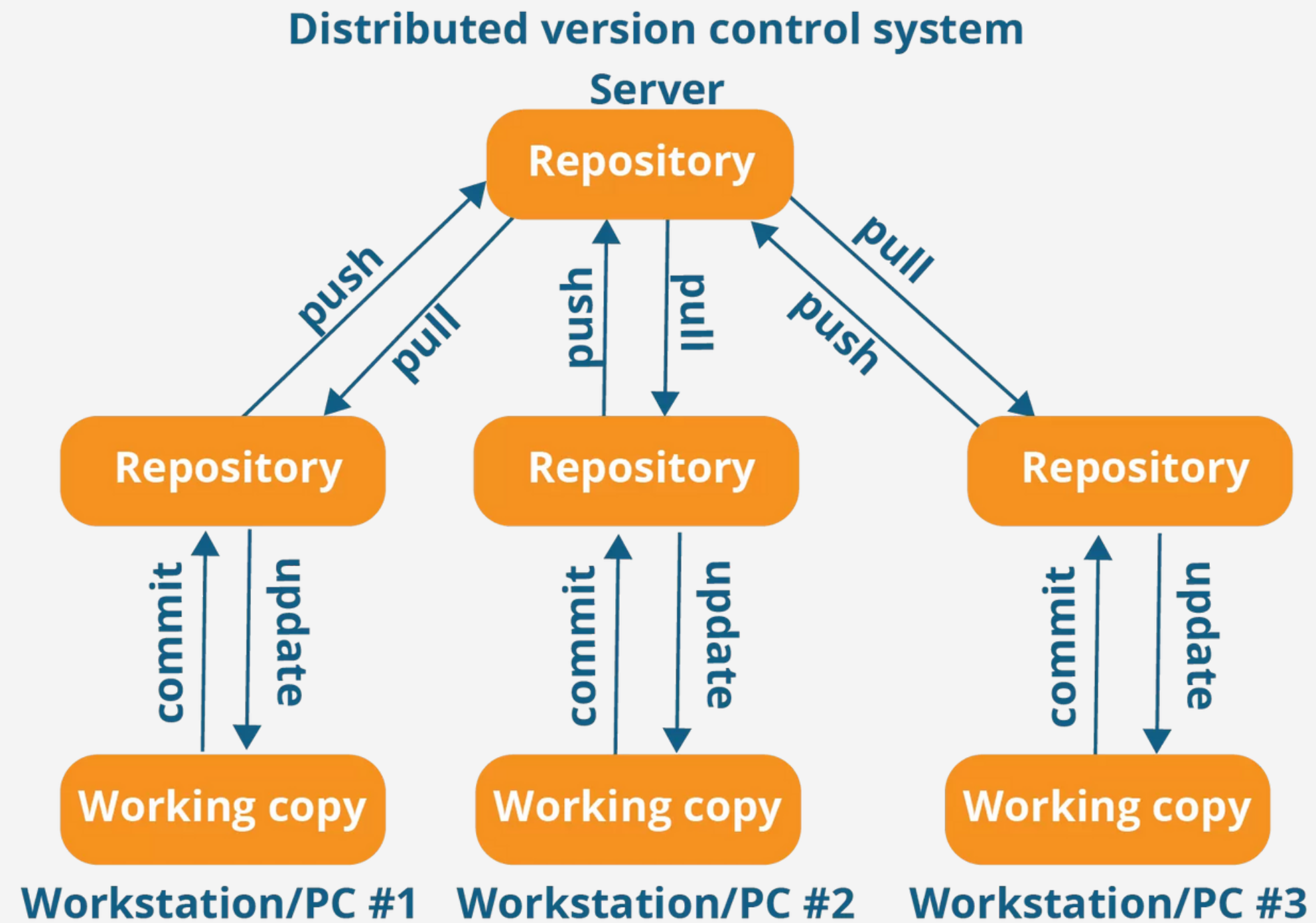
10 Comandos de Git Que Todo Desarrollador Debería Saber

Git es una parte importante de quien programa a diario (especialmente si estás trabajando con un equipo) y se usa extensamente en la industria de software. Desde que existe una gran variedad de comandos que puedes utilizar, dominar Git requiere tiempo. Pero algunos comandos se utilizan más...

 freeCodeCamp.org / Jan 11, 2021

Colaboración entre varios usuarios

10



Ejemplo Práctico

Utilizando git y realizando un programa de gestión de empleados

```
no usages
4 ▶ public class Main {
5
6     7 usages
    public static String[] empleados = new String[5];
    1 usage
7     public static int[] horasTrabajadas = new int[5];
8
9     no usages
    public static String[] nombreBonos = new String[5];
    no usages
10    public static int[] porcentajesBonos = new int[5];
11
12    1 usage
    public static void crearEmpleado() {
13        Scanner input = new Scanner(System.in);
14        System.out.println("Ingrese el nombre del nuevo empleado: ");
15        String nombre = input.nextLine();
16        System.out.println("Ingrese las horas trabajadas: ");
17        int horas = input.nextInt();
18
19        // validar que no existe ese empleado
```

¿Dudas o
preguntas?



Tarea #1



Realizar una breve investigación sobre los parámetros de una función por referencia y parámetros por valor,

¿Qué son? Y un ejemplo corto en Java, en un documento PDF con el nombre: [IPC1]T1_#Carne.pdf

Asistencia

- Formulario de Asistencia:
-

