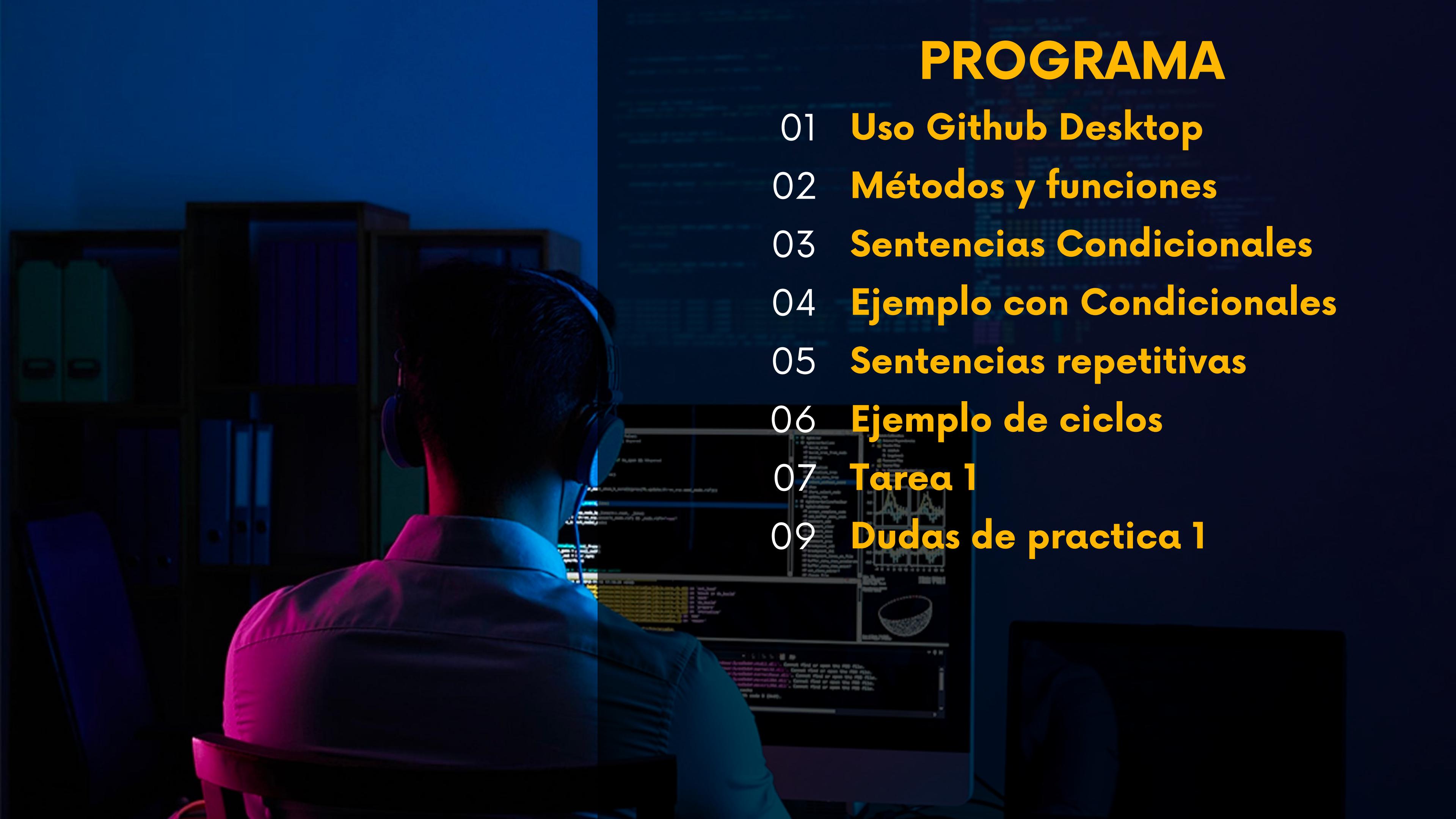


IPC1 SECCIÓN F

Clase 3

Freddy Monterroso



PROGRAMA

- 01 **Uso Github Desktop**
- 02 **Métodos y funciones**
- 03 **Sentencias Condicionales**
- 04 **Ejemplo con Condicionales**
- 05 **Sentencias repetitivas**
- 06 **Ejemplo de ciclos**
- 07 **Tarea 1**
- 09 **Dudas de practica 1**

CREACIÓN DE REPOSITORIO EN GITHUB

- 1. Crear una cuenta en github.com**
- 2. Iniciar sesión**
- 3. Crear un nuevo repositorio**

The screenshot shows the GitHub homepage with a dark theme. At the top, there is a search bar with the placeholder "Search or jump to...". To the right of the search bar are links for "Pull requests", "Issues", and "Codes". Below the search bar, there is a "Top Repositories" section featuring a "New" button. A search input field with the placeholder "Find a repository..." is also present. On the left side of the main content area, there is a list of repositories:

- freddy-201212854/BD2-Grupo_4
- freddy-201212854/so1_proyecto_g10
- crisborr8/practica2_G1
- freddy-
- freddy-201212854/machingelearnig_20121285

On the right side of the page, there is a promotional sidebar for "GitHub Global" with the text: "Join GitHub Global. Prepare for a career in tech. Get the practical industry events, learning resources, and more." There are also sections for "Popular offers you have not claimed" and "Hub to get help with hackathons & events".

4. Asignarle un nombre al repositorio.

5. En el caso de las practicas y proyectos colocar los repositorios como privados.

6. Seleccionar la opción "Add a README file".

7. Click en "Crear repositorio".

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner * Repository name *

 freddy-201212854 /

Great repository names are short and memorable. Need inspiration? How about [didactic-pancake](#)?

Description (optional)

 Public
Anyone on the internet can see this repository. You choose who can commit.

 Private
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: None ▾

Choose a license

INVITAR A MIEMBROS A UN REPOSITORIO

- 1. Click en settings**
- 2. Click en Collaborators**
- 3. Deberá de ingresar su contraseña**
- 4. Presionar el botón "Add people"**
- 5. Ingresar el usuario o correo que desea que participe en su repositorio.**

The screenshot shows the GitHub repository settings interface. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The Settings tab is active, indicated by a red underline. On the left, a sidebar lists General, Access (Collaborators is selected), Code and automation (Branches, Tags, Actions, Webhooks, Codespaces, Pages), and Security (Code security and analysis, Deploy keys). The main content area is titled 'Who has access' and shows that it's a PRIVATE REPOSITORY where Only those with access to this repository can view it. It also indicates DIRECT ACCESS with 3 have access to this repository. Below this, a 'Manage' button leads to a 'Manage access' modal. The modal features a 'Select all' checkbox, a search bar labeled 'Find a collaborator...', and a list of collaborators. One collaborator, '201212808', is listed with a 'Remove' button next to their name.

Code Issues Pull requests Actions Projects Security Insights Settings

General

Access

Collaborators

Code and automation

Branches

Tags

Actions

Webhooks

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

DIRECT ACCESS

3 have access to this repository. 3 collaborators.

Manage

Add people

Select all

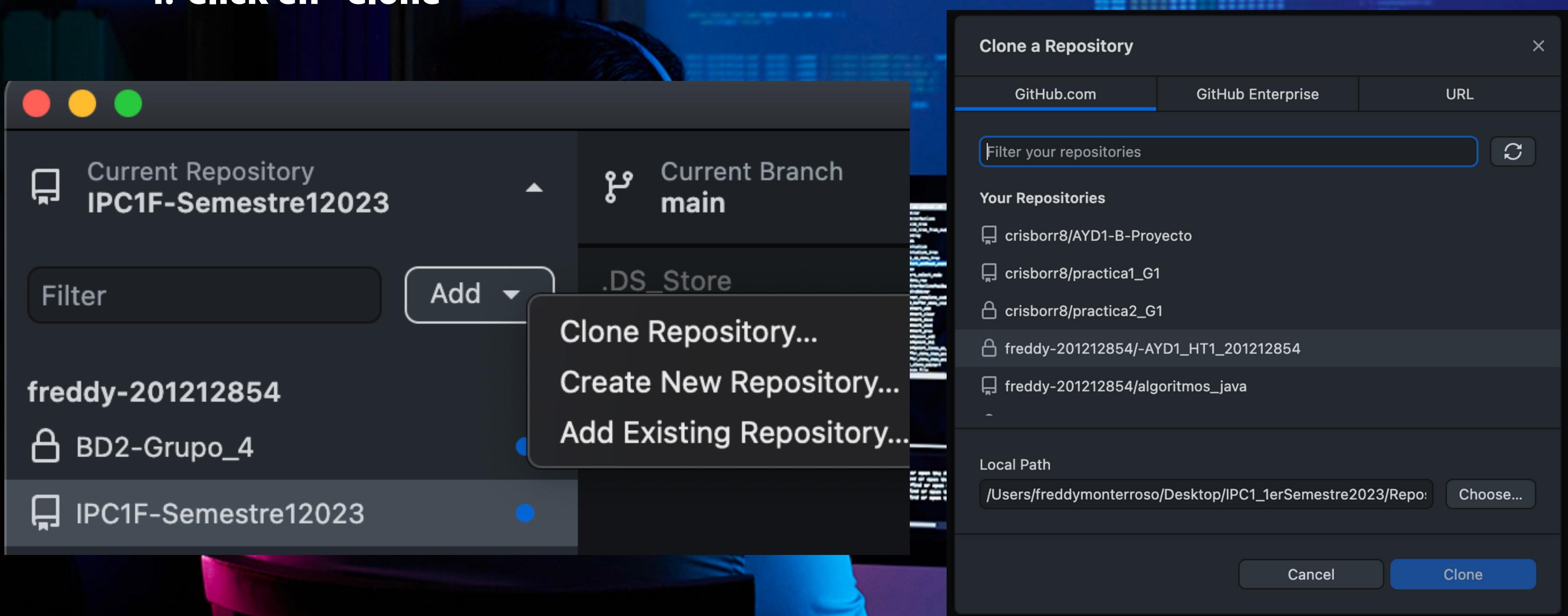
Find a collaborator...

201212808
Collaborator

Remove

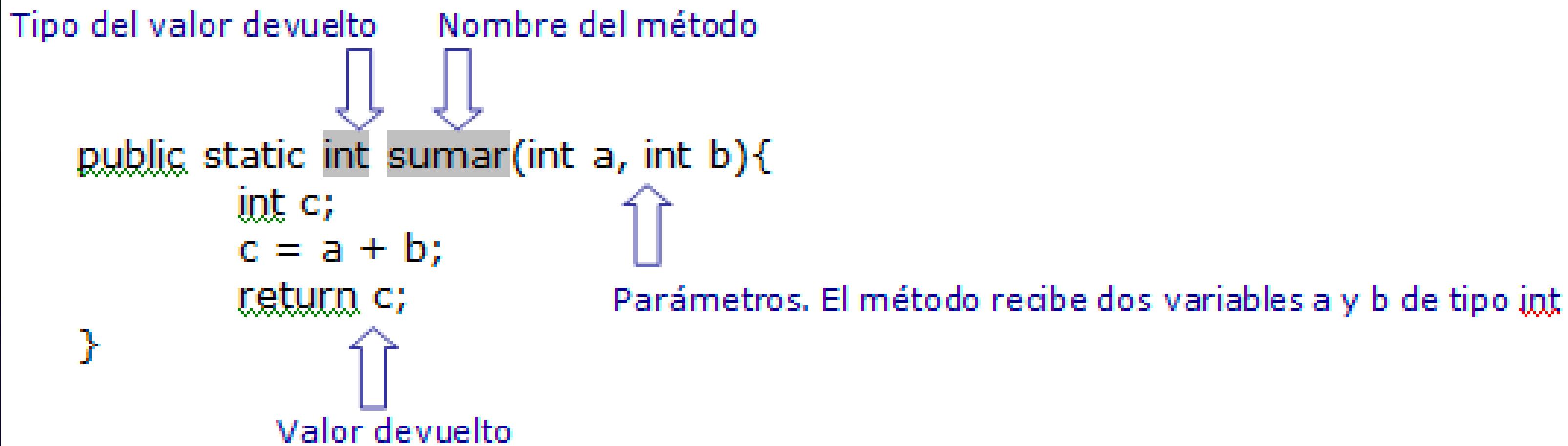
CLONAR UN REPOSITORIO CON GITHUB DESKTOP

1. Clic en "Add" y seleccionar clonar repositorio.
2. Buscar y seleccionar el repositorio a clonar.
3. Seleccionar la ubicación del repositorio.
4. Click en "Clone"



METODOS Y FUNCIONES

Bloques de código que se puede reutilizar, en el caso de las funciones estas si retornan un valor y los métodos no.



SENTENCIAS CONDICIONALES

Estas instrucciones se ejecutan en base a una proposición, y esta compuesta por valores, operadores de comparación y lógicos.

IF

```
if ( condicion ) {  
    // bloque de instrucciones  
}
```

IF-ELSE

```
if (condicion) {  
    // bloque de instrucciones  
} else {  
    // bloque de instrucciones  
}
```

IF-ELSE IF -ELSE

```
if (condicion) {  
    // bloque de instrucciones  
} else if (condicion){  
    // bloque de instrucciones  
} else if (condicion){  
    // bloque de instrucciones  
} else {  
    // bloque de instrucciones  
}
```

OPERADORES CONDICIONALES

Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
<code>==</code>	igual que	<code>7 == 38</code>	<code>false</code>
<code>!=</code>	distinto que	<code>'a' != 'k'</code>	<code>true</code>
<code><</code>	menor que	<code>'G' < 'B'</code>	<code>false</code>
<code>></code>	mayor que	<code>'b' > 'a'</code>	<code>true</code>
<code><=</code>	menor o igual que	<code>7.5 <= 7.38</code>	<code>false</code>
<code>>=</code>	mayor o igual que	<code>38 >= 7</code>	<code>true</code>

OPERADORES LÓGICOS

OPERADOR	NOMBRE	EJEMPLO	DEVUELVE VERDADERO CUANDO...
<code>&&</code>	y	<code>(7 > 2) && (2 < 4)</code>	las dos condiciones son verdaderas
<code> </code>	o	<code>(7 > 2) (2 < 4)</code>	al menos una de las condiciones es verdadera
<code>!</code>	no	<code>!(7 > 2)</code>	la condición es falsa

SENTENCIAS CONDICIONALES

Switch

```
switch(valor) {  
    case valor:  
        // bloque de instrucciones  
        break;  
  
    case valor2:  
        // bloque de instrucciones  
        break;  
  
    default:  
        // bloque de instrucciones  
        break;  
}
```



PRIMER EJEMPLO - CONDICIONES

Realice un menú con las siguientes opciones y sus respectivas funcionalidades:

- 1. Realice un programa que permita el ingreso de un número y que indique por medio de la consola si es positivo, negativo o neutro (utilice un método).**
- 2. Imprimir si un número ingresado es par o impar (utilice una función).**
- 3. Realice un programa que permita el ingreso de la nota de un curso y que por medio de la consola indique si el estudiante a Aprobado o Reprobado (utilice un método).**
- 4. Imprimir si un usuario o contraseña son validas (utilice una función).**
- 5. Ingresar e imprimir un listado de productos con su respectivo nombre y precio utilizando una matriz (utilice un método).**
- 6. Salirse de la aplicación**

SENTENCIAS REPETITIVAS O CICLOS

Permiten ejecutar un bloque de instrucciones varias veces.

```
for (inicio; condicion; incremento) {  
    // bloque de instrucciones  
}
```

```
While(condicion) {  
    // bloque de instrucciones  
}
```

```
do {  
    // bloque de instrucciones  
} while(condicion);
```



PRIMER EJEMPLO - CICLOS

- 1. Realizar un contador del 1 al 10 utilizando los 3 ciclos vistos.**
- 2. Convertir el menú de opciones del ejemplo anterior en repetitivo y que nunca finalice a menos que se ingresa la opción "Salir".**
- 3. Mostrar los numeros pares del 1 al 100**
- 4. Mostrar los números impares del 1 al 100**



¿DUDAS?

