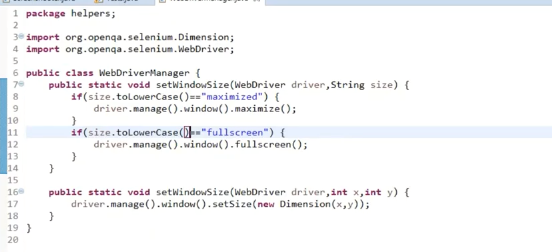
# Ordenando Codigo

# Clase captura.

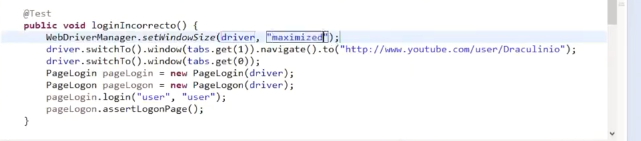
## clase static

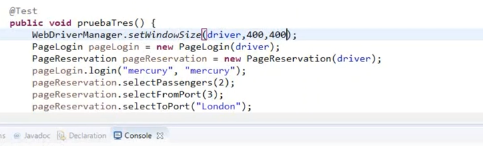
nos evita crear objetos para usar los métodos / no volver a reiniciar el objeto desde cero ej: las listas( cargar una lista infinita veces y no tener que volver a iniciar cada vez que creo el objeto)

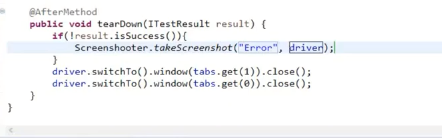












# Clase Maneje tamaño pantallas

Sobrecarga de métodos: tienen la misma asignación (nombre) pero diferentes parámetros

# Codigo

## Clase captura

**public** **static** **void** sacarCaptura(String nombre,WebDriver driver)

{

File captura=((TakesScreenshot)driver).getScreenshotAs(OutputType.***FILE***);

**try** {

FileUtils.*copyFile*(captura,**new** File(nombre+" "+System.*currentTimeMillis*()+".png"));

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

## Clase WebdriverGestion

**public** **static** **void** setWindowsSize(WebDriver driver,String tamanio)

{

**if**(tamanio.toLowerCase()=="maximizar")

{

driver.manage().window().maximize();

}

**if**(tamanio.toLowerCase()=="pantallaFull")

{

driver.manage().window().fullscreen();//pantalla completa

}

}

**public** **static** **void** setWindowsSize(WebDriver driver, **int** ancho,**int** alto)

{

driver.manage().window().setSize(**new** Dimension(ancho,alto));//tamaño

}

## Test

**public** **void** loginIncorrecto()//login incorrecto

{

WebDriverGestion.*setWindowsSize*(driver,"maximizar");

@Test

**public** **void** PruebaTres()

{

WebDriverGestion.*setWindowsSize*(driver,200,300);

@AfterMethod

**public** **void** tearDown1(ITestResult result)

{

Capturas.*sacarCaptura*( "metodo captura", driver);