

Revisión y Validación de conocimientos

La práctica tendrá la finalidad de conocer y reforzar los conceptos teóricos y prácticos estudiados durante las Clases:

- ✓ **Maven:** Comandos básicos
- ✓ **Eclipse:** Crear Proyectos, paquetes, Clases, funciones principales del Entorno de Desarrollo.
- ✓ **Git y GitHub:** Comando Básicos, resolución de conflicto, repositorios Locales, Híbridos, on Cloud.
- ✓ **Junit:** Testing Unitario, Métodos básicos, crear test y obtener análisis.
- ✓ **Java:** Clases, objetos, atributos, Archivos, Colecciones, Tipos de Datos, control de flujo(Bucles y Ctrl. Selectivo). Métodos de Wrappers y Clases String, Integer, Arrays, Array List, etc. Excepciones.

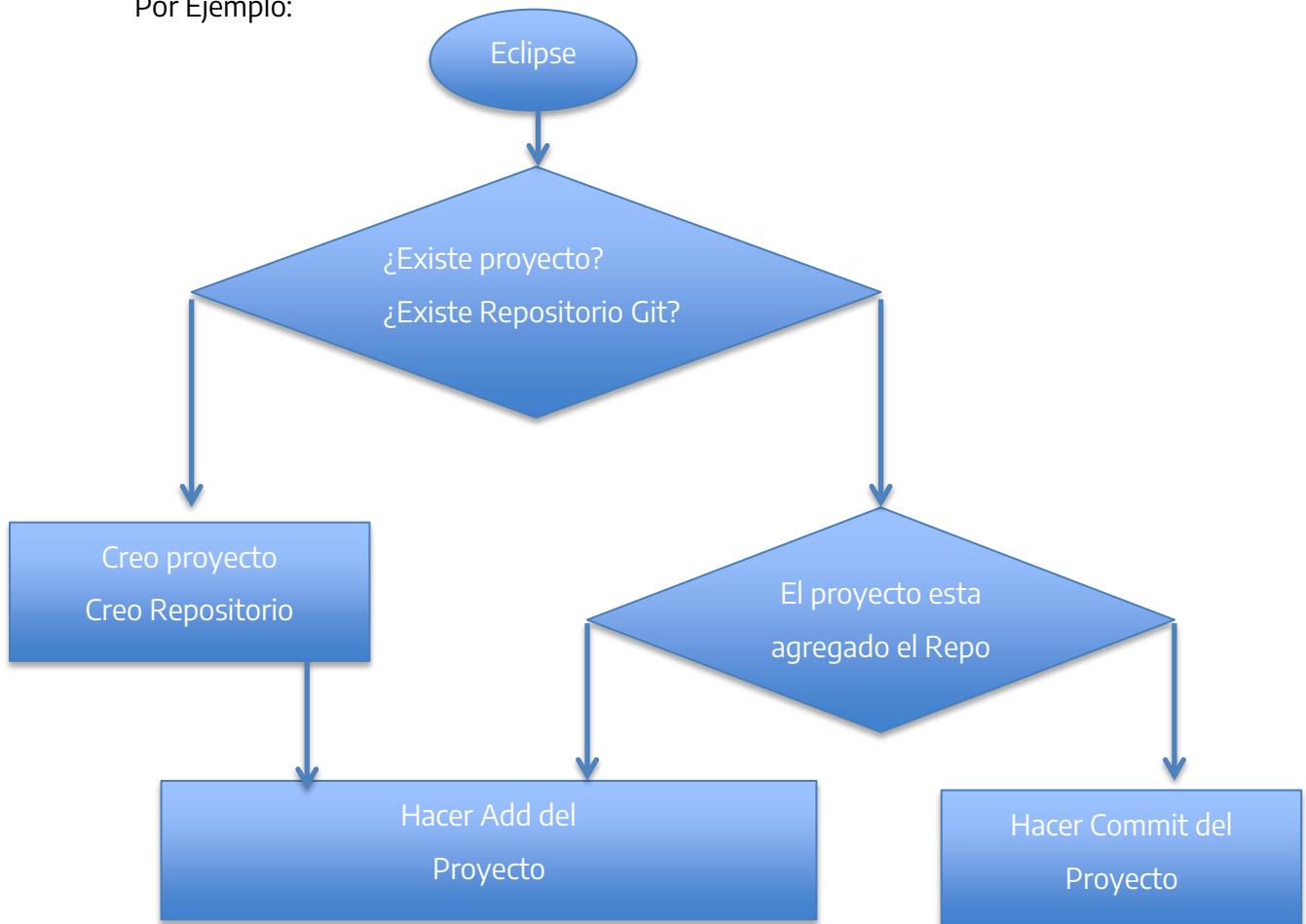
Maven:

1. El comando “mvn install”, ¿para qué sirve?
2. ¿El archivo principal de configuración de un proyecto gestionado por Maven tiene extensión XML? ¿Cómo es su estructura?
3. Liste los comandos básicos de Maven para Elipse. Describa brevemente cada uno.

Git - GitHub

1. Liste los comandos básicos de Git. Describa brevemente cada uno.
2. Realice un diagrama de flujo donde muestre como usaría los comandos de Git para subir un proyecto desde Eclipse a GitHub.

Por Ejemplo:



Excepciones:

Analice el código siguiente:

```
public class Main {  
  
    public static void main(String[] args) {  
        String opcion = "Si";  
        reservar(opcion);  
        System.out.println("turno fue reservada correctamente");  
    }  
    static String reservar(boolean opcion){  
        if(opcion) return "Si";  
        else return "No";  
    }  
}
```

1. ¿Funciona o se produce una excepción?
2. Explique narrativamente la mecánica en caso que funcione o Explique donde y porque se la excepción en caso que no funcione.
3. En cualquier caso: Agregue los bloques de control Try-Catch para atrapar las excepciones y muestre un mensaje adecuado por pantalla.
4. Si tengo el siguiente código y en la 2da línea se arroja una Exception, describir que sucede.

```
1 try{  
2     habitacion.reservar()  
3 } catch (Exception ex) {}  
4 System.out.println("La habitación fue reservada correctamente")
```

Colecciones - clase HashMap

1. Liste los métodos para agregar, leer y borrar un objeto HashMap.
2. Responda:
Si se tiene un objeto xx de clase HashMap<String, Producto>
a. La clave es un producto y el valor un texto
b. xx.put("algo",137) no debería compilar
c. xx.get("algo") si la clave existe debe retornar un objeto tipo producto o null
d. xx.get("algo") si la clave existe debe retornar un objeto tipo String

Java – Literales, caracteres especiales, palabras reservadas

1. De los siguientes caracteres (<, -, @, |, /, //, \, \\, #, i, !, {, [, (,),], }, >), cual son reservados o no se deberían usar por ser carácter de escape.
2. Describa brevemente en qué casos de usan y muestre un ejemplo.

Java – Métodos

1. ¿LocalTime.now() es un método o variable?
2. LocalTime.now() , de que Clase, Interfaz o Librería proviene.

Java – Archivos, bucles de control, Métodos

1. Dados el archivo y el código:

```
1;40;*jabón en polvo
*3;10;esponjas
2;*100;chocolates
```

```
public class Main {

    public static void main(String[] args) throws IOException {
        String archivo = "C:\\\\Entrada.txt";
        for (String linea : Files.readAllLines(Paths.get(archivo))) {

            if (linea.startsWith("*")) {
                System.out.println(linea);
            }
        }
    }
}
```

Explicar narrativamente lo que hace el código e indicar cuál sería la salida por pantalla.

2. Identificar todos los Métodos usados y que a Clase pertenecen, describa brevemente.
3. Si usted está en Windows, y ejecuta un programa de Java desde C:\\Users\\SuNombre\\ y quiere abrir el archivo C:\\Users\\SuNombre\\Documentos\\unTexto.txt. Marque la opción correcta:

- a. Para abrir el archivo puede usar en el String la ruta "unText.txt"
- b. Para abrir el archivo puede usar en el String la ruta "C:\\Users\\SuNombre\\Documentos\\unTexto.txt"
- c. Para abrir el archivo puede usar en el String la ruta "Documentos\\unText.txt"
- d. b y c son correctas
- e. todas son incorrectas

Java – Operadores lógicos y Bucles

1. El dados el operador && (AND o "Y") y el || (OR o "O"), y las variables a = 17, b = 4 y c = 4. Indique el resultado de las siguientes operaciones.

- a. (a > b) && (c != b):
- b. (a > b) && ((c * b) < a):
- c. (a == c) || (b == c):

2. Dado el siguiente código que imprime por pantalla?

```
public class Main {  
  
    public static void main(String[] args) throws IOException {  
        int a = 9;  
        int b = 1;  
        while (a > (b-1)) {  
            if (a < b*3) {  
                a = a + b;  
            }  
            a = a - b*2;  
            System.out.println(a);  
        }  
    }  
}
```

Java – Arrays

1. Si tenemos el siguiente array: `int numeros[] = new int[]{1,37,16};`
¿Qué valor es `numeros [2]`?

Junit – Unit Test

1. Liste los métodos básicos de un Test unitario (assertEqual, assertTrue, etc.), explique brevemente.
2. Liste los identificadores básicos de Test unitario (@Test, @Before, etc.), explique brevemente.
3. Si suponemos que el atributo “tieneHambre” del objeto “hogar” es falso, entonces:
 - a. El test que tiene la sentencia Assertions.assertTrue (!hogar. tieneHambre) no pasa satisfactoriamente.
 - b. El test que tiene la sentencia Assertions.assertEquals (false, hogar. tieneHambre) pasa satisfactoriamente
 - c. El test que tiene la sentencia Assertions.assertTrue (hogar. tieneHambre) no pasa satisfactoriamente.
 - d. El test que tiene la sentencia Assertions.assertFalse (hogar. tieneHambre) no pasa satisfactoriamente.

Responda: ----

Java – POO

1. ¿La palabra reservada “this” tiene sentido en un método “static”?
2. ¿Una clase puede heredar de varias clases pero implementar una sola interfaz?
3. Dada la siguiente clase y el objeto “carrito1”, indique la opción correcta:

```
class CarritoCompra {  
    private List<ItemCarrito> items;  
    public CarritoCompra() {}  
    public void agregarItem(ItemCarrito ic) {  
        this.items.add(ic);  
    }  
    } .....  
CarritoCompra carrito1 = new CarritoCompra();
```

- a. El carrito de compra no puede ser instanciado sin parámetros
 - b. Para funcionar, en la declaración, en lugar de usar `List<ItemCarrito>` se debe usar `ArrayList<ItemCarrito>`
 - c. `carrito1.agregarItem(unItem)` agregara un item a la colección `items`
 - d. `carrito1.agregarItem(unItem)` arrojará un error
-
- 4. ¿Main() que tipo de método es? ¿Para qué sirve?
 - 5. Describa los modificadores de acceso (público, privado, protegido)
 - 6. ¿Cómo podemos crear los getter and setter de forma casi automáticas en Eclipse? ¿Para qué sirven?
 - 7. ¿Cómo instancio una Clase?
 - 8. ¿Cómo heredo de una Clase Abstracta?
 - 9. ¿Cómo heredo una interfaz?
 - 10. ¿Cómo utilizo los métodos de una clase?
 - 11. ¿Cómo utilizo los métodos de una clase abstracta?
 - 12. ¿Cómo utilizo los métodos de una Interfaz?

Eclipse

- 1. Describa los pasos para crear un proyecto en Eclipse
- 2. Describa los pasos para crear una clase principal con el método `Main()`, dentro del proyecto.
- 3. Describa los pasos para crear paquetes
- 4. Describa los pasos para crear una instancia de una clase C dentro de un paquete X, desde otra Clase A que está en otro paquete Z

