

1.-Definiciones:

- **Variable:** Cuando reservamos un espacio en memoria para poder guardar un valor y poder declararlo próximamente.
- **Constante:** Tipo de propiedad de un método
- **Herencia:** Como su propio nombre indica, una clase(hijo) hereda las propiedades y métodos de una clase abstracta(padre).
- **Clase:** Una plantilla que tiene métodos y propiedades, un ejemplo sería los planos de una casa.
- **Excepción:** Capturamos un error utilizando try-catch. Con el try lanzamos el error y con el catch lo capturamos.
- **Arquitectura del Software MVC:** Nos permite distinguir las distintas partes/funcionalidades de un programa.

2: Diferencia entre for y while(con ejemplo):

- **For:** Indicamos una condición que se debe cumplir, luego el objetivo, y al final la condición;

```
for (i = 2; i < 50; i++){  
•  
• }
```

- **While:** Se repetirá hasta que cumpla la condición

```
• int i = 1;  
•  
• while (i<=10) {  
•     System.out.println(i++);  
• }
```

3: Clase abstracta e interfaz: La clase abstracta contiene un método vacío/incompleto.

- **Diferencias:**
 - La interfaz no tiene métodos/propiedades completas y la clase abstracta por el contrario sí.
 - Para poder usar una clase abstracta se utiliza “Extend”, y para un método “Implements”
- **Similitudes**
 - Contienen métodos incompletos (name ())

4: Escribe un fragmento de código en java que utilice una estructura if-else para determinar si un número es positivo, negativo o cero

```
public static void main(String[] args) {
    System.out.println("Introduce un número: ");
    int num = Keyboard.nextInt();

    if(num > 0){
        System.out.println("Es positivo");
    }else(num < 0){
        System.out.println("Es negativo");
    }else{
        System.out.println("Es cero");
    }

    System.out.println("El número introducido es: ");
}

public class Keyboard {
    private static Scanner scanner = new Scanner(System.in);

    public static String text() {
        return scanner.nextInt();
    }
}
```

5: Fragmento de código que utilice un bloque try-catch para manejar una posible excepción al dividir dos números.

```
int num1;
int num2;

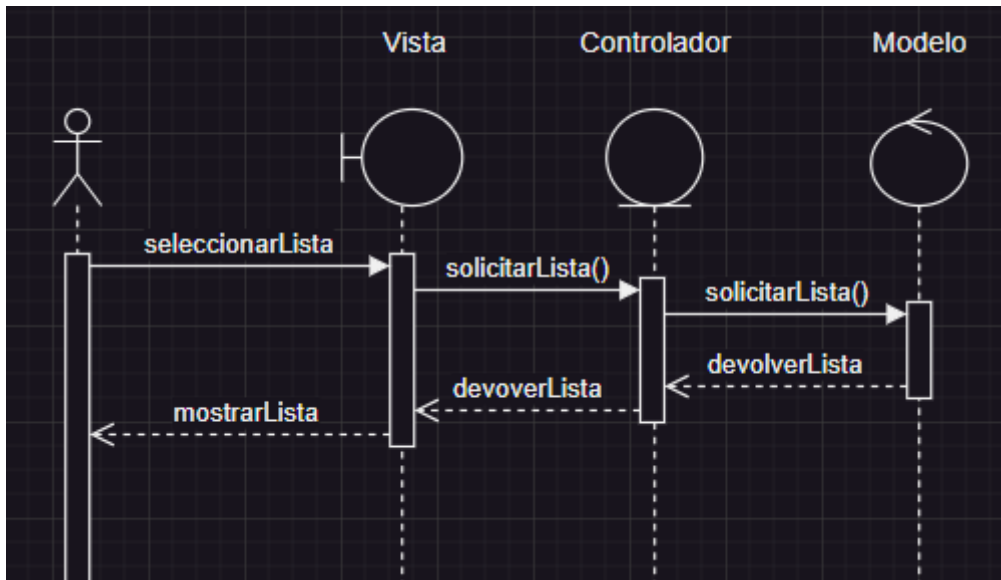
try{
    System.out.println("Sumando: "+ num1 "y " + num2 + "nos da: " +
num1*num2);
}catch(Exception e) {
    System.out.println("Fallo rey " + e);
}
```

6: ¿Qué ocurre si no se captura una excepción?

Nos devuelve la posición en memoria.

7: Describe como interactúan el Modelo, Vista y Controlador en una aplicación MVC.

- El **modelo** es quien guarda la información y manipula, el **controlador** quien hace de intermediario entre la vista y el modelo, y la **vista** una interfaz donde muestra y recibe la entrada(teclado)



8: Escribe el código para que la clase Perro herede de la clase Animal

- 8.1. Utilice el constructor de la clase padre para recibir el nombre del animal y en el toString tiene que imprimir el nombre del animal y el ruido que hace (0.5 puntos):
- 8.2. Añade un método solo del Perro y un ejemplo en el que no se pueda usar y un caso en el que si se pueda usar (1 punto):

