

Erick Antillón

Introducción a la programación y computación 1

Clase 8 - Clases abstractas e Hilos

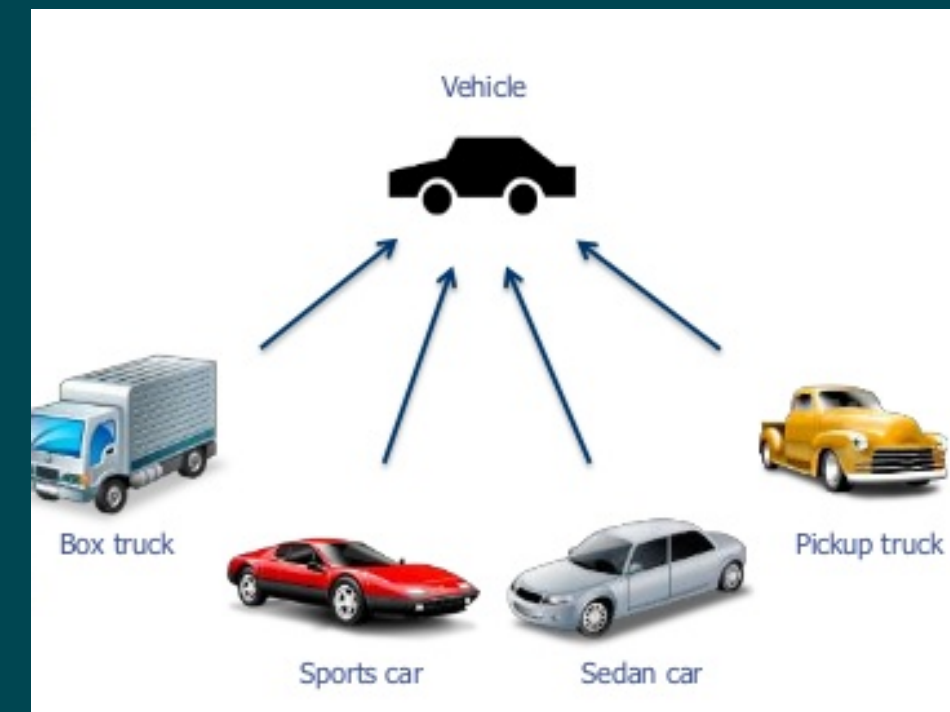


Agenda

- Abstracción
- Clases Abstractas
- Interfaces
- Hilos (Thread y Runnable)
- Hilos con interfaz gráfica
- Eventos del teclado
- Asistencia

Abstracción

- Es un concepto fundamental en la Programación Orientada a Objetos (POO). Se refiere a la capacidad de definir conceptos complejos en términos de sus características más importantes o relevantes, mientras se ocultan los detalles de implementación.
- Se trata de simplificar la representación de un objeto para que sea más fácil de entender y utilizar.
- En Java, la abstracción se implementa a través de interfaces y clases abstractas, las cuales proporcionan un nivel de abstracción por encima de las clases concretas.
- También facilita la comprensión del código y la reutilización de componentes. Además, al utilizar abstracciones en lugar de clases concretas, se pueden hacer cambios en la implementación sin afectar a los demás clientes que utilizan la abstracción.



Clases Abstractas

- Las clases abstractas son un tipo de clase en Java que no pueden ser instanciadas, sino que se utilizan como plantillas para definir otras clases.
- Puede contener métodos abstractos (al menos uno de sus métodos debe ser abstracto), que son métodos sin implementación, así como métodos, que si tienen una implementación completa.
- Las clases abstractas se utilizan comúnmente como una forma de proporcionar una implementación parcial o genérica de una funcionalidad, y para obligar a las clases hijas a proporcionar una implementación específica de esa funcionalidad.
- Sus niveles de visualización deben ser o public o protected (nunca private).

```
abstract class Animal {  
    public abstract void animalSound();  
    public void sleep() {  
        System.out.println("Zzz");  
    }  
}
```

Interfaces

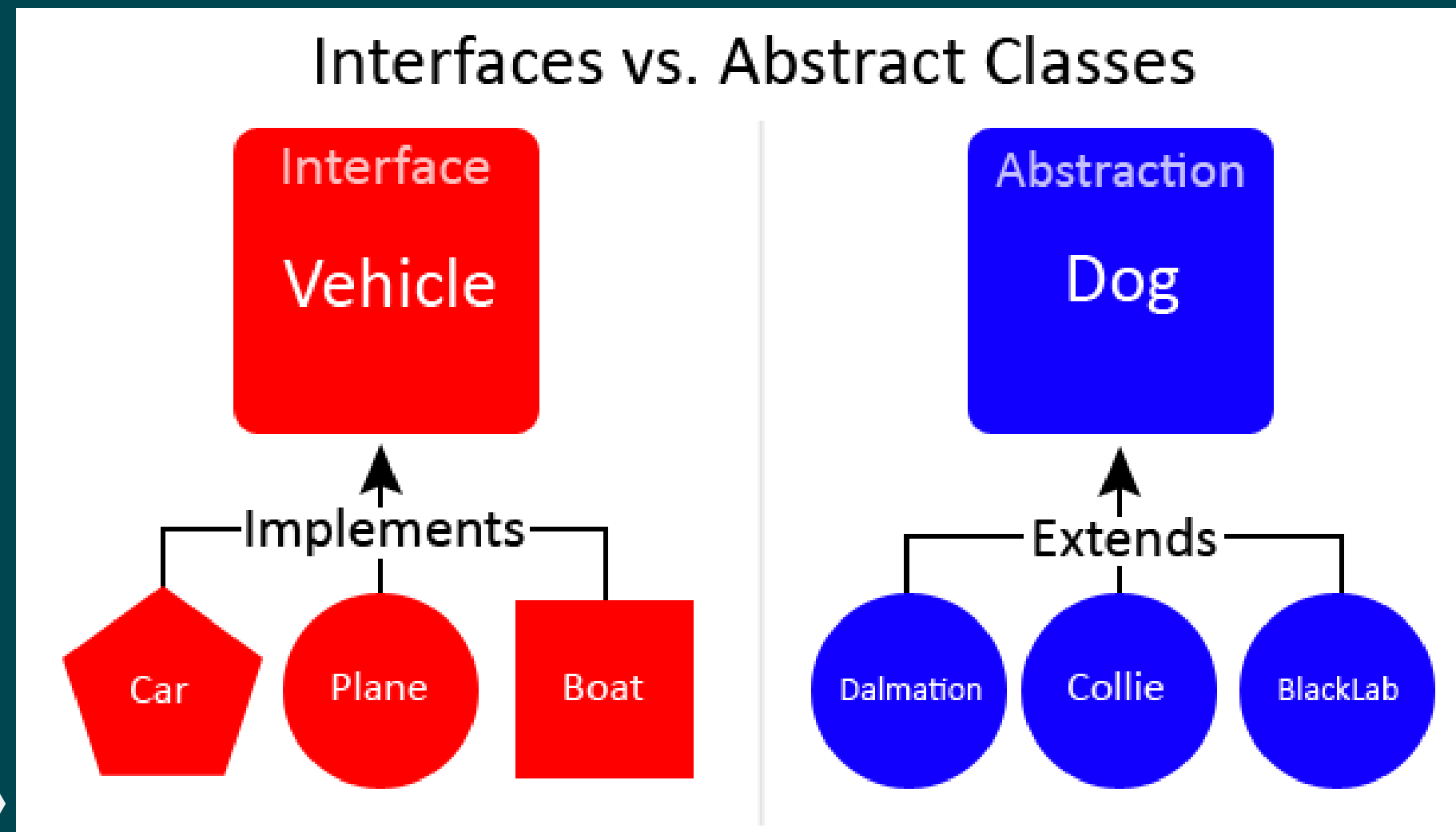
- Es una colección de métodos abstractos y constantes que se utiliza para definir un contrato o conjunto de comportamientos que una clase debe implementar.
- Las interfaces no pueden contener atributos de instancia ni constructores. Solo propiedades (atributos) CONSTANTES.
- Se utilizan para establecer un contrato entre varias clases que tienen un comportamiento similar, pero no necesariamente comparten una jerarquía de herencia. Una clase que implementa una interfaz implementar todos los métodos definidos en la interfaz.
- Solo puede tener métodos con acceso público (no pueden ser protected o private)



```
public class MiClase implements Interfaz1, Interfaz2 {  
  
}
```

Interfaz vs Clase Abstracta

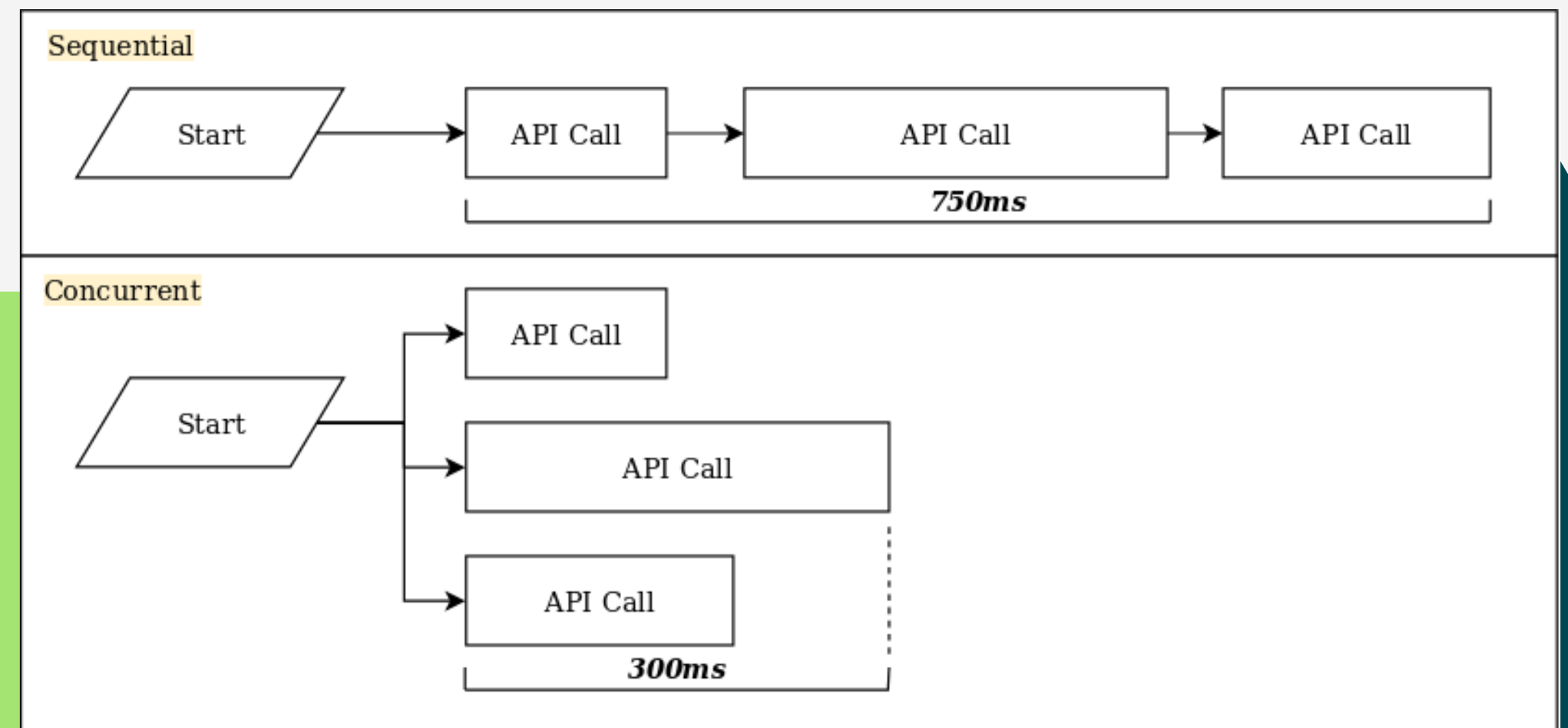
- Una clase puede implementar varias interfaces, pero solo puede heredar de una clase abstracta



Hilos en Java

Un hilo (también conocido como thread) es una unidad de ejecución concurrente dentro de un programa. Cada hilo es una secuencia separada de ejecución que puede ejecutarse en paralelo con otros hilos dentro del mismo programa

Los hilos en Java se crean a través de la clase **Thread**. Cada hilo tiene su propia pila de ejecución y su propio estado de ejecución, pero comparten el mismo espacio de memoria y recursos del sistema operativo que otros hilos.

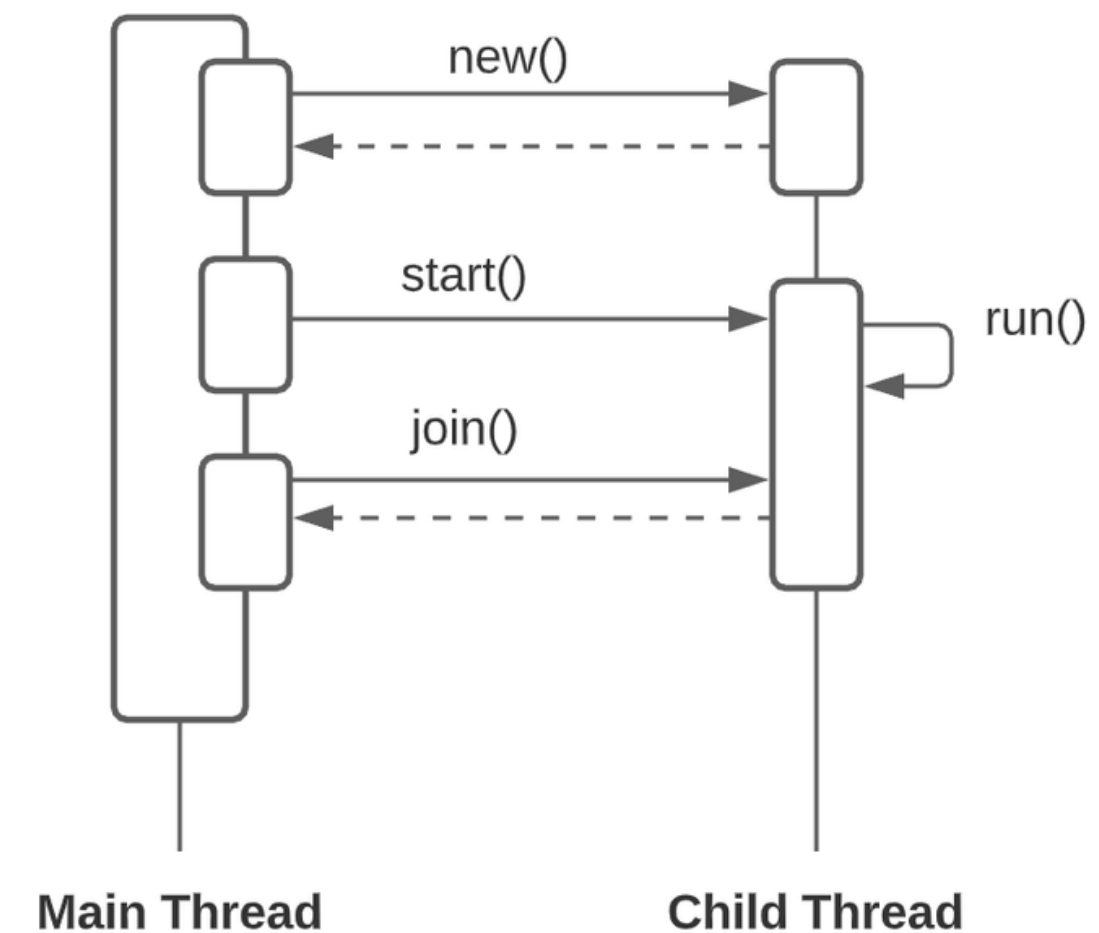


Hilos en Java

- Los hilos en Java constan de tres partes principales:
- Runnable: Es el código que se ejecutará en el hilo. Puede implementarse mediante la interfaz *Runnable* o extendiendo la clase *Thread*.
- Inicialización: Para crear un hilo en Java, es necesario crear una instancia de la clase *Thread* y pasarle como parámetro una instancia de la clase *Runnable* que represente el código a ejecutar en el hilo.
- Ejecución: Una vez que se ha creado un hilo, se puede iniciar su ejecución llamando al método *start()*. El método *start()* crea un nuevo hilo y llama al método *run()* del objeto *Runnable* proporcionado como parámetro.

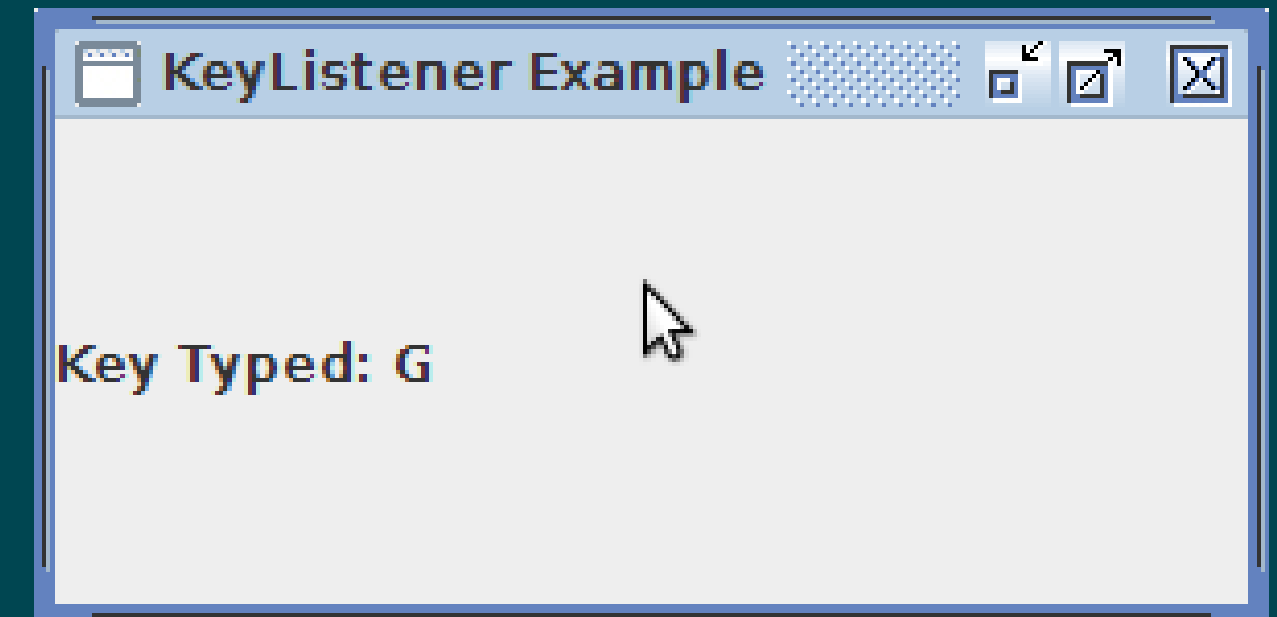
Hilos con Interfaz grafica

- Cuando se trabaja con hilos en una interfaz gráfica, es importante tener en cuenta que las actualizaciones en la interfaz deben realizarse en el hilo de eventos de la interfaz gráfica (EventDispatchThread). Esto significa que cualquier actualización de la interfaz gráfica debe hacerse en el hilo de eventos y no en el hilo que realiza el trabajo pesado.
- *SwingUtilities.invokeLater();*



Eventos del teclado

- En Swing, para manejar los eventos del teclado, se necesita crear un objeto que implemente la interfaz `KeyListener` y registrar ese objeto como un listener, que define tres métodos que se llaman cuando ocurren ciertos eventos de teclado: *`keyPressed()`*, *`keyReleased()`*, y *`keyTyped()`*.
- Se pueden registrar `KeyListeners` tanto en `JFrame` como en cualquier componente que extienda la clase `java.awt.Component` en Java, como `JPanel`, `JButton`, `TextField`, entre otros.



¿Dudas o
preguntas?



Asistencia

- Formulario de Asistencia:
<https://forms.gle/FQKaopZ6zJisMHLr8>

