

*Universidad Internacional del Ecuador*



*Ingeniería en software*

*Nombre:*

*Daniela Paillacho*

*Asignatura:*

*Lógica de programación*

*Tema:*

*Diseño de Diagramas de funcionalidad y arquitectura previo al  
desarrollo de un software.*

*Fecha de envió: 04/11/2024*

*Fecha de entrega 13/11/2024*

*Quito-Ecuador*

## **1. Tipos de Diagramas de funcionalidad y arquitectura de aplicaciones que existen**

### **1.1 Tipos de Diagramas de Funcionalidad**

#### **Diagrama de Flujo:**

Representa el flujo de control de un proceso o sistema mediante formas y flechas.

Útil para simplificar procesos complejos y visualizar pasos secuenciales.

#### **Diagrama de Casos de Uso:**

Muestra las interacciones entre los usuarios (actores) y el sistema.

Destaca los requisitos funcionales y las funcionalidades del sistema.

#### **Diagrama de Actividad:**

Describe el flujo de actividades en un proceso de negocio.

Detalla las tareas, decisiones y flujos de trabajo.

#### **Diagrama de Secuencia:**

Representa la interacción entre objetos en el orden en que ocurren.

Útil para detallar el comportamiento temporal de un sistema.

### **1.2 Tipos de Diagramas de Arquitectura de Aplicaciones**

#### **Arquitectura Monolítica:**

Todo el sistema se desarrolla como una única unidad de código.

Puede dificultar la escalabilidad y el mantenimiento, pero es más simple de implementar.

#### **Arquitectura de Microservicios:**

El sistema se divide en servicios pequeños y autónomos que se comunican a través de APIs.

Mejora la escalabilidad, la flexibilidad y la capacidad de actualización independiente.

#### **Arquitectura Cliente-Servidor:**

El cliente solicita servicios al servidor, que los procesa y devuelve la respuesta.

Común en aplicaciones web y de bases de datos.

#### **Arquitectura en Capas:**

El sistema se organiza en capas funcionales, como presentación, lógica de negocio y acceso a datos.

Facilita la separación de preocupaciones y el mantenimiento del código.

## *Piedra, papel o tijera.*

### ➤ *Resolución de problemas*

✚ Jugador 1 y Jugador 2 (o la computadora) seleccionan una de las tres opciones: piedra, papel o tijera.

✚ Evaluar las opciones elegidas siguiendo las reglas del juego:

- Piedra vence a tijera.
- Tijera vence a papel.
- Papel vence a piedra.
- Si ambos jugadores eligen la misma opción, el resultado es un empate.

✚ Según las reglas:

- Si Jugador 1 elige piedra y Jugador 2 elige tijera, gana Jugador 1.
- Si Jugador 1 elige tijera y Jugador 2 elige papel, gana Jugador 1.
- Si Jugador 1 elige papel y Jugador 2 elige piedra, gana Jugador 1.
- Si las elecciones son diferentes, pero no se cumple ninguna de las condiciones anteriores, gana Jugador 2.

✚ Anunciar al ganador, o en caso de empate, declarar "Empate".

### ➤ *Diseño de funcionalidades*

Para diseñar el juego de Piedra, Papel o Tijera a detalle, podemos utilizar diferentes diagramas que nos ayudarán a visualizar y planificar las funcionalidades del software.

- Diagrama de Casos de Uso

Este diagrama muestra las interacciones entre los usuarios (actores) y el sistema, destacando los requisitos funcionales.

✚ Actores:

Jugador

Sistema (Computadora)

✚ Casos de Uso:

Iniciar juego

Seleccionar opción (piedra, papel, tijera)

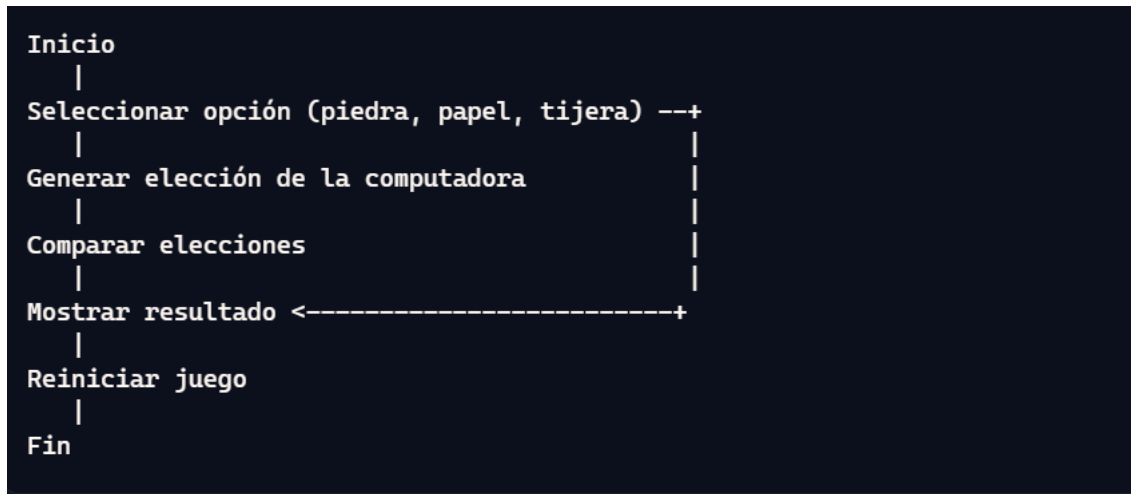
Generar elección de la computadora

Comparar elecciones

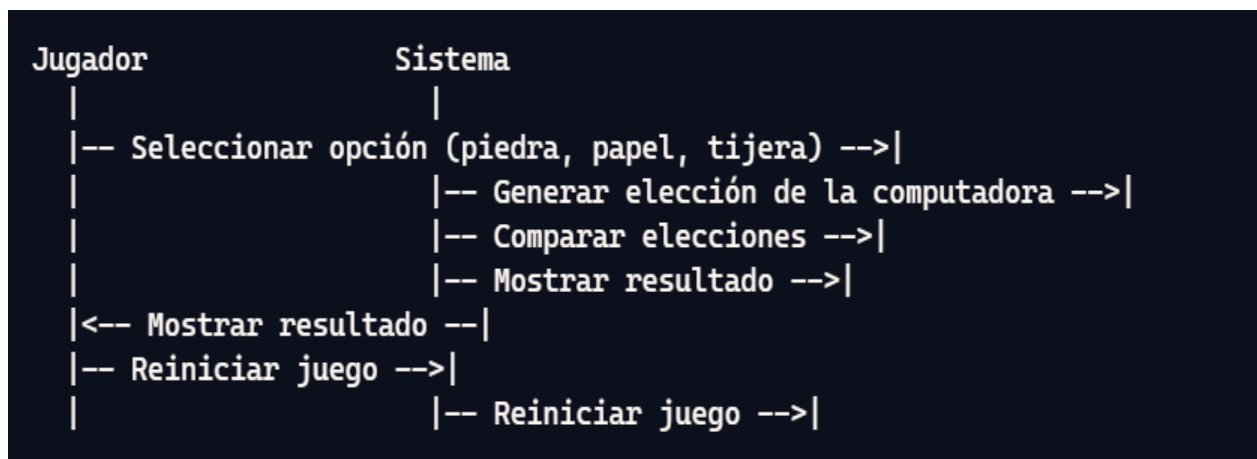
Mostrar resultado

Reiniciar juego

➤ *Diagrama de Actividad*



➤ *Diagrama de Secuencia*



➤ *Diagrama de Clases*

Este diagrama muestra la estructura del sistema en términos de clases y sus relaciones.

🚦 **Juego**

- Atributos: estadoDelJuego, elecciónJugador, elecciónComputadora, resultado
- Métodos: iniciarJuego(), seleccionarOpción(), generarElecciónComputadora(), compararElecciones(), mostrarResultado(), reiniciarJuego()

### ➤ *Funcionalidades del Software*

#### ○ **Iniciar Juego:**

El sistema inicia y muestra un menú inicial con opciones para empezar el juego.

#### ○ **Seleccionar Opción:**

El jugador puede seleccionar una de las tres opciones: piedra, papel o tijera.

El sistema registra la elección del jugador.

#### ○ **Generar Elección de la Computadora:**

El sistema genera aleatoriamente una elección para la computadora.

#### ○ **Comparar Elecciones:**

El sistema compara la elección del jugador y la elección de la computadora.

Determina el resultado (ganar, perder o empate).

#### ○ **Mostrar Resultado:**

El sistema muestra el resultado del juego al jugador.

Se muestra un mensaje indicando si el jugador ganó, perdió o empató.

#### ○ **Reiniciar Juego:**

El sistema ofrece la opción de jugar de nuevo o terminar el juego.

Si el jugador elige jugar de nuevo, se reinicia el juego.

### ➤ *Diagrama de arquitectura*

[https://miro.com/app/embed/uXjVLG\\_smVY=?pres=1&frameId=3458764606552822470&embedId=972458613302](https://miro.com/app/embed/uXjVLG_smVY=?pres=1&frameId=3458764606552822470&embedId=972458613302)