

**UNIVERSIDAD INTERNACIONAL DEL ECUADOR**

**UIDE – QUITO – 2025**

**LOGICA DE PROGRAMACION**

## **TRABAJO AUTÓNOMO**

**Elaborado por:**

Michael Alexander Lescano Moya

**FECHA:**

18 de mayo de 2025

**TEMA:**

Selección del Programa a desarrollar / Generación de Diagramas funcionales y Arquitectura de  
Software

## Diagrama de Casos de Uso – Juego del Ahorcado

El diagrama de casos de uso representa las interacciones entre los actores (usuarios) y las funcionalidades del sistema.

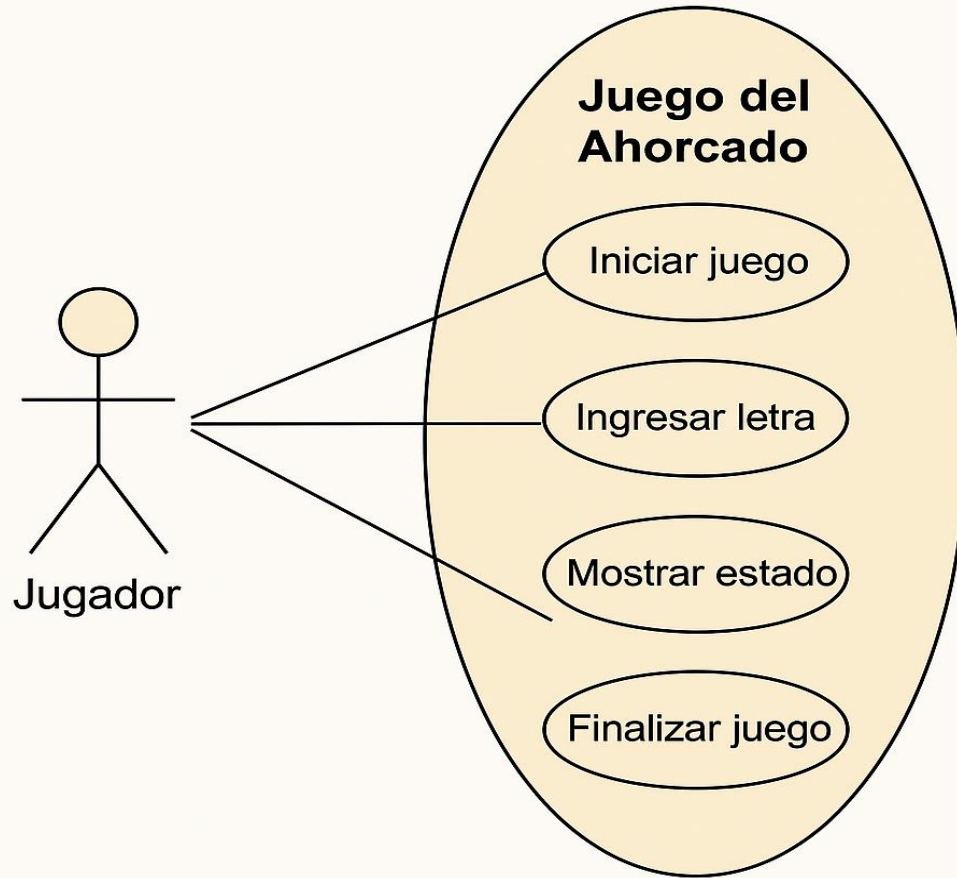
### Actores:

- **Jugador:** Usuario que interactúa con el juego.

### Casos de Uso:

- **Iniciar juego:** Comienza una nueva partida.
- **Seleccionar dificultad:** Elige el nivel de dificultad.
- **Adivinar letra:** Introduce una letra para adivinar la palabra secreta.
- **Verificar letra:** El sistema comprueba si la letra está en la palabra.
- **Actualizar estado del juego:** Muestra el progreso actual, incluyendo letras acertadas y partes del ahorcado dibujadas.
- **Finalizar juego:** Concluye la partida, indicando si el jugador ganó o perdió.

Este diagrama ayuda a visualizar las funcionalidades principales que el sistema debe ofrecer al usuario.



## Diagrama de Arquitectura de Software – Juego del Ahorcado

El diagrama de arquitectura de software muestra la estructura general del sistema, incluyendo los componentes principales y sus interacciones.

### Componentes Principales:

- **Interfaz de Usuario (UI):** Encargada de la interacción con el jugador.
- **Lógica del Juego:** Contiene las reglas y el flujo del juego.
- **Gestor de Palabras:** Maneja la selección y almacenamiento de palabras secretas.
- **Base de Datos (opcional):** Almacena estadísticas, palabras y configuraciones.

### Flujo de Datos:

1. El jugador interactúa con la UI para iniciar el juego y realizar acciones.
2. La UI comunica las acciones a la Lógica del Juego.
3. La Lógica del Juego solicita palabras al Gestor de Palabras.
4. El Gestor de Palabras puede acceder a la Base de Datos para obtener palabras.
5. La Lógica del Juego actualiza el estado y envía la información a la UI para mostrarla al jugador.

Este diagrama proporciona una visión clara de cómo se organiza el sistema internamente y cómo los componentes interactúan entre sí.

## Arquitectura de Aplicación

