Universidad Internacional del Ecuador (UIDE)

Ciencias de la Computación

LOGICA DE PROGRAMACIÓN

Título: Trabajo Autónomo 1

Docente: ESTEFANIA VANESSA HEREDIA JIMENEZ.

Johan Steven Quinapallo Tituaña

11TH NOVIEMBRE, 2025

Introducción

En el desarrollo de software, antes de programar una solución es necesario comprender claramente el problema, definir sus elementos funcionales y diseñar una arquitectura adecuada. Este proceso permite anticipar la estructura lógica del sistema, identificar posibles errores y asegurar que el producto final responda correctamente a los requerimientos planteados.

El presente trabajo desarrolla el análisis y el diseño previo del juego del Ahorcado, utilizando un diagrama de flujo funcional y un diagrama de arquitectura en capas, con el objetivo de planificar de manera clara cómo funcionará el sistema. Ambos diagramas permiten visualizar la secuencia de acciones del juego, sus decisiones principales y cómo se organiza el software internamente antes de su implementación.

Además, este enfoque facilita una comprensión más profunda del funcionamiento del programa, ya que conecta conceptos teóricos y prácticos del diseño de software. Al representar el comportamiento del juego paso a paso y dividir su estructura en capas, se logra una visión más ordenada y profesional del proyecto, lo que constituye una base sólida para el desarrollo posterior del código.

Desarrollo

2. Investigación de Tipos de Diagramas

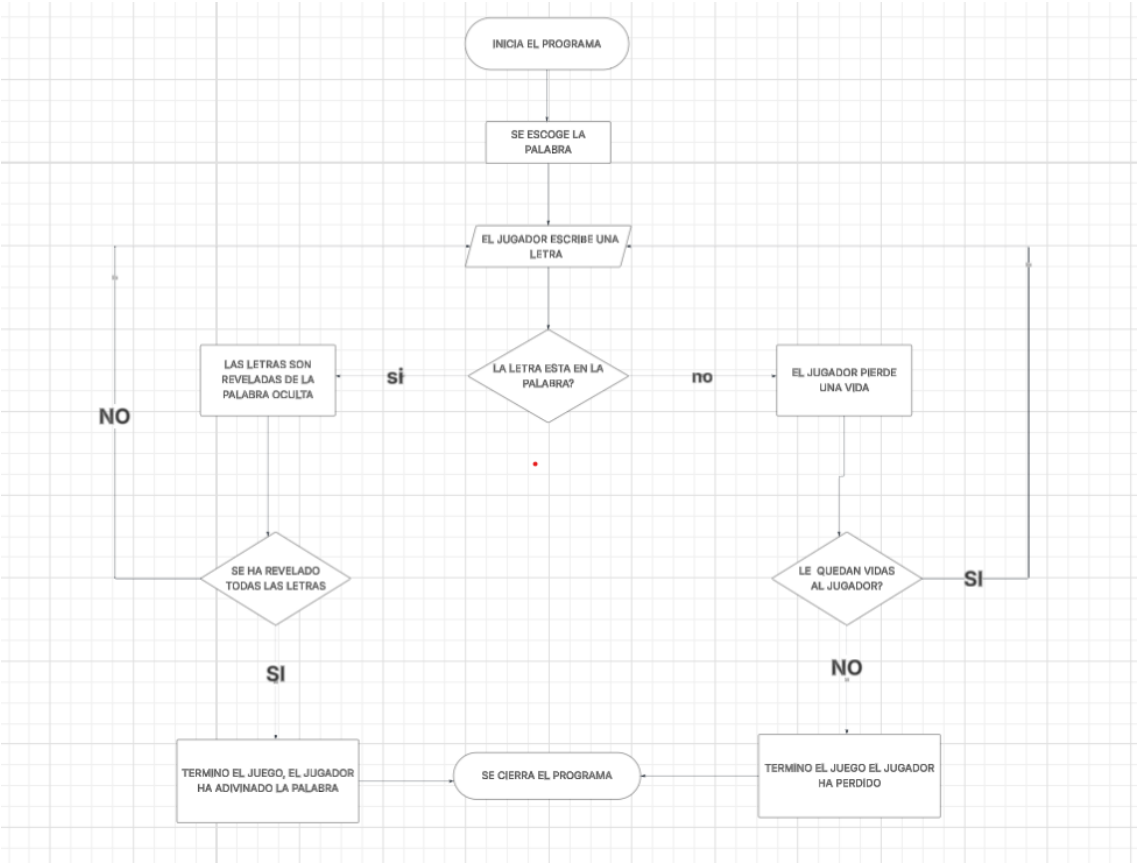
Para esta actividad se investigaron diferentes tipos de diagramas utilizados en la ingeniería de software.

2.1. Diagramas funcionales

Los diagramas funcionales permiten representar el comportamiento dinámico de un sistema, es decir, cómo fluye la información, qué decisiones se evalúan y qué procesos se ejecutan. Dentro de esta categoría existen varios tipos relevantes:

Tipos De Diagramas	
Diagrama De Flujo	El diagrama de flujo es una representación gráfica secuencial que muestra los pasos que sigue un proceso. Utiliza símbolos estandarizados como óvalos (inicio/fin), rombos (decisiones), rectángulos (procesos) y paralelogramos (entradas y salidas). Es útil para visualizar de manera clara la lógica del sistema y sus ciclos repetitivos (Sommerville, 2016).
Diagrama De Casos De Uso	Este diagrama, perteneciente al lenguaje UML, describe las interacciones entre los usuarios (actores) y el sistema. Permite identificar qué funcionalidades debe ofrecer el software desde la perspectiva del usuario, y es una herramienta clave para el levantamiento de requerimientos (Pressman & Maxim, 2020).
Diagrama De Actividades	Representa el flujo de actividades dentro de un proceso. Muestra acciones, decisiones, concurrencias y rutas alternativas. Es especialmente útil para detallar procesos complejos que requieren múltiples caminos posibles (Satzinger, Jackson & Burd, 2016).
Diagrama De Estados	Se utiliza para describir los diferentes estados por los que pasa un objeto dentro de un sistema y los eventos que provocan los cambios entre esos estados. Es ideal para modelar sistemas reactivos como juegos o dispositivos interactivos (Larman, 2017).

Para este proyecto se eligió el **diagrama de flujo**, ya que representa de manera clara el funcionamiento del juego del Ahorcado. En él se muestran los pasos que el sistema sigue desde la selección de la palabra hasta que el usuario gana o pierde. La lógica del diagrama incluye la comprobación de letras, actualización del ahorcado y repetición del ciclo hasta llegar a un resultado final.

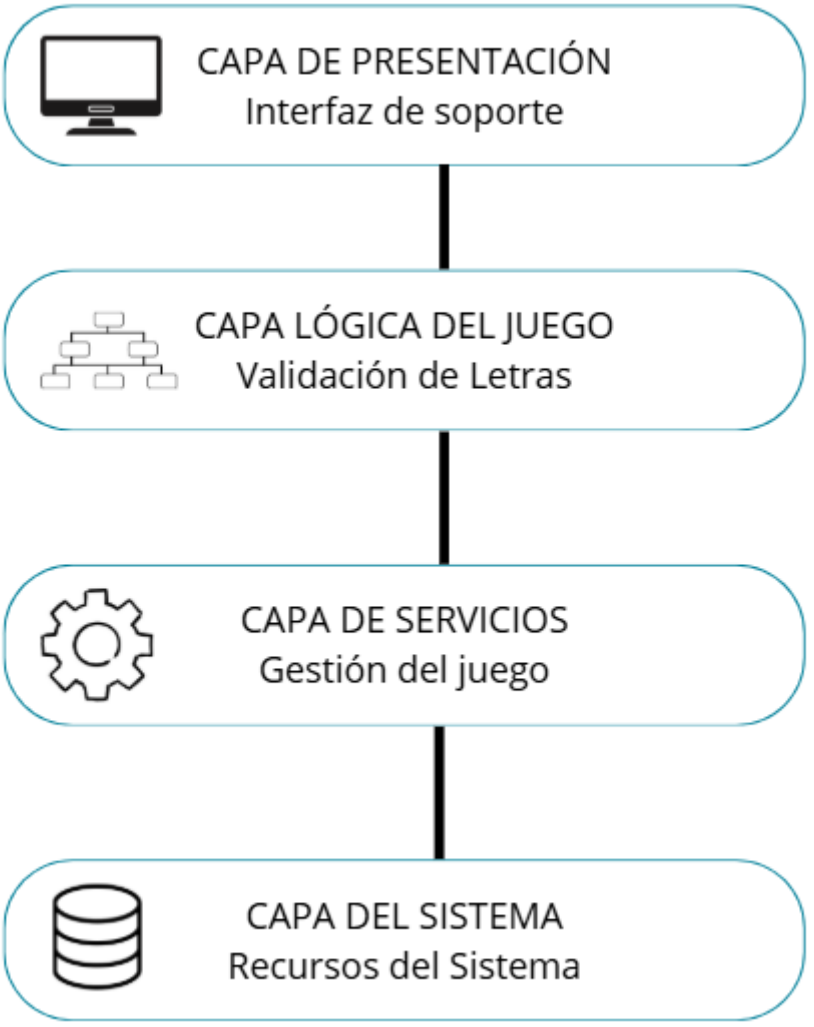


2.2 Diagramas de arquitectura

Los diagramas de arquitectura representan la organización interna del software, mostrando cómo se dividen y relacionan sus componentes principales. Este tipo de modelado permite planificar la estructura del sistema antes de comenzar la programación.

Arquitectura En Capas	Divide el sistema en niveles jerárquicos, donde cada capa cumple una función específica. Por ejemplo: interfaz de usuario, lógica de negocio, servicios y sistema base. Esta arquitectura facilita el mantenimiento, la reutilización y la separación de responsabilidades (Bass, Clements & Kazman, 2013).
Diagrama De Componentes	Representa los módulos que conforman un sistema y las dependencias entre ellos. Es útil para visualizar cómo se ensamblan las diferentes partes del software a nivel estructural.
Arquitectura Cliente-Servidor	Modelo donde el cliente solicita servicios y el servidor los provee. Es común en aplicaciones web, redes y sistemas distribuidos.
Arquitectura MVC	Modelo que divide la aplicación en tres partes: Modelo (datos), Vista (interfaz) y Controlador (lógica). Es ampliamente utilizado en aplicaciones gráficas e interfaces modernas

Para este trabajo se eligió la **Arquitectura en Capas**, debido a que permite organizar de forma clara los elementos del juego del Ahorcado, separando la lógica de validación, la interfaz visual y los servicios internos.


CAPA DE PRESENTACIÓN
Interfaz de soporte
CAPA LÓGICA DEL JUEGO
Validación de Letras
CAPA DE SERVICIOS
Gestión del juego
CAPA DEL SISTEMA
Recursos del Sistema

3. Análisis del problema

El juego del Ahorcado consiste en adivinar una palabra oculta proporcionando letras.

Cada vez que el usuario se equivoca, se dibuja una parte del ahorcado.

El juego termina cuando:

- El jugador completa la palabra → Gana
- El ahorcado se completa por errores acumulados → Pierde

Entradas del usuario

Letras proporcionadas durante el juego.

Salidas del sistema

- Representación de los espacios en blanco.
- Estado de la palabra parcialmente descubierta.
- Dibujo del ahorcado.
- Estado final (“Ganaste” o “Perdiste”).

Reglas principales

Cada acierto revela posiciones en la palabra.

Cada error dibuja una parte adicional del ahorcado.

El ciclo se repite hasta ganar o perder.

Conclusión

El proceso de análisis y diseño previo a la programación es fundamental para comprender correctamente un problema y planificar una solución eficiente. El diagrama de flujo muestra visualmente el ciclo del juego del Ahorcado, mientras que la arquitectura en capas organiza el software de manera clara.

Este trabajo permite entender cómo funcionará el juego antes de implementarlo, cumpliendo con las buenas prácticas del desarrollo de software.

URL DE VIDEO:
https://drive.google.com/file/d/1FSuiHTjhsvSWH6fBVCLzSO7B1fIF42yE/view?usp=sharing

Bibliografía

Universidad Internacional de La Rioja. (2023). ***Tipos de diagramas: Qué son y para qué sirven*****.**
https://ecuador.unir.net/actualidad-unir/tipos-diagramas/

Senn, J. (2004). ***Análisis y diseño de sistemas de información*** **(2.ª ed.). McGraw-Hill.**
https://www.mheducation.com/

Pressman, R. (2010). ***Ingeniería del software: Un enfoque práctico*** **(7.ª ed.). McGraw-Hill.**
https://www.mheducation.com.mx/ingenieria-del-software-un-enfoque-practico-7ed.html

MindManager. (2023). ***Diagrama funcional: Qué es y cómo se usa*****.**
https://www.mindmanager.com/en/features/functional-chart/