

UNIVERSIDAD INTERNACIONAL DEL ECUADOR

Facultad de Ciencias Técnicas

Ingeniería en Ciberseguridad

Docente

ING. PAULINA VIZCAINO

Estudiante

ARTURO GUEVARA NARVÁEZ

Actividad:

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

Aprendizaje Autónomo 1.

1.- Nombre del Software a desarrollar.

El juego escogido es “**El Ahorcado**”, el cual tiene un origen muy antiguo, probablemente de la época Victoriana (siglo XIX), y se popularizó como un juego de lápiz y papel.

2.- Investigación y selección del Diagrama.

Diagrama de funcionalidad (UML)

Se utilizará para este caso el **Diagrama de Funcionalidad de Flujo**, que representa gráficamente al algoritmo o proceso. Utiliza símbolos estandarizados conectados por flechas para mostrar el orden lógico de las operaciones que se deben realizar para resolver un problema.

Diagrama de Arquitectura de Software

Se escogió el **Diagrama de Arquitectura de Software por Capas**; que consta de dividir el algoritmo en capas, con la intención de que cada capa tenga un rol muy definido. Todas se colocan de forma horizontal, de tal forma que cada capa solo puede comunicarse con la capa que está inmediatamente por debajo, por lo que, si una capa quiere comunicarse con otras que están mucho más abajo, tendrán que hacerlo mediante la capa que está inmediatamente por debajo. En este caso lo organizamos por **Capa de Interfaz de Usuario**, **Capa Lógica de Negocio (BLL)**, **Capa de Datos del Juego**.

3.- Análisis del problema.

Juego:

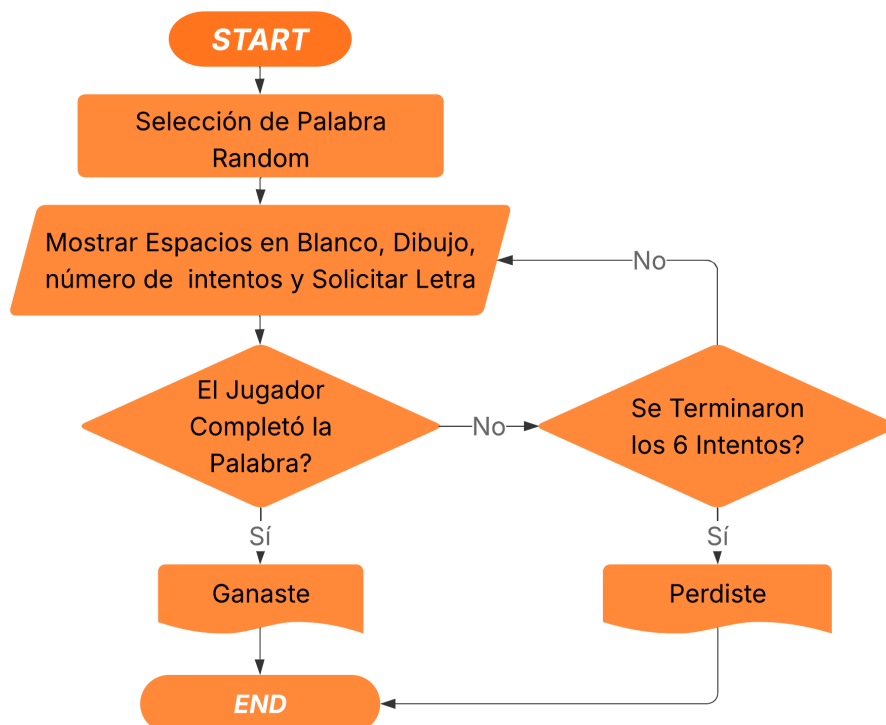
El ahorcado es un juego de un participante donde debe adivinar la palabra random secreta que ha sido seleccionada por la computadora. Donde el algoritmo permite al usuario escoger una a una las letras que conforman la palabra secreta con un máximo de 6 errores.

Requisitos funcionales:

- Seleccionar una palabra secreta Random.
- Mostrar los espacios en blanco “_” que corresponden a cada una de las letras de la palabra secreta.
- Mostrar al Usuario el “grafico” de estado del “ahorcado”.
- Mostrar el número de intentos que le restan.
- Mostrar al usuario las letras que ya han sido ingresadas.
- Recibir la selección del jugador.
- Validar el número de aciertos y errores para determinar si el fin del juego es una victoria o derrota.

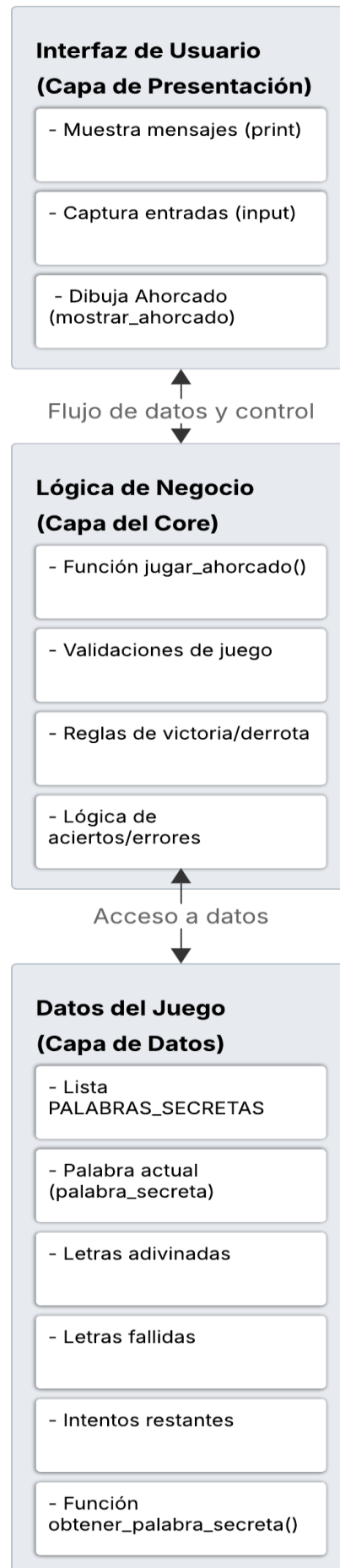


4.- Diagrama funcional (Diagrama de Flujo)



Fuente: Elaboración propia

5.- Diagrama de Arquitectura del software.



Fuente: Elaboración propia

6.- Algoritmo Básico (pseudocódigo)

INICIO

```
// 1. Inicialización
ELEGIR_PALABRA_SECRETA_AL_AZAR
INICIAR_PALABRA_OCULTA_CON_GUIONES (ej. "_ _ _")
ESTABLECER_INTENTOS_MAXIMOS = 6
INICIAR_LETRAS_ADIVINADAS_POR_JUGADOR = lista vacía
INICIAR_LETRAS_FALLIDAS_POR_JUGADOR = lista vacía

// 2. Bucle principal del juego
MIENTRAS INTENTOS_RESTANTES > 0 Y PALABRA_OCULTA CONTIENE GUIONES:

    // 2.1. Mostrar estado actual
    MOSTRAR_DIBUJO_AHORCADO (basado en INTENTOS_FALLIDOS)
    MOSTRAR_PALABRA_OCULTA
    MOSTRAR_INTENTOS_RESTANTES
    MOSTRAR_LETRAS_FALLIDAS

    // 2.2. Pedir y validar entrada
    PEDIR_LETRA_AL_JUGADOR
    MIENTRAS LETRA_INVALIDA (no es una sola letra, ya fue
    adivinada/fallida):
        MOSTRAR_MENSAJE_ERROR_ENTRADA
        PEDIR_LETRA_AL_JUGADOR_DE_NUEVO
    FIN MIENTRAS

    // 2.3. Procesar la letra
    SI LETRA_ELEGIDA ESTA_EN PALABRA_SECRETA:
        ACTUALIZAR_PALABRA_OCULTA (revelar letra en todas sus
        posiciones)
        MOSTRAR_MENSAJE_LETRA_CORRECTA
    SINO:
        INCREMENTAR_INTENTOS_FALLIDOS
        AÑADIR_LETRA_A_LETRAS_FALLIDAS
        REDUCIR_INTENTOS_RESTANTES_EN_UNO
        MOSTRAR_MENSAJE_LETRA_INCORRECTA
    FIN SI
FIN MIENTRAS

// 3. Fin del juego
SI PALABRA_OCULTA NO_CONTIENE_GUIONES:
    MOSTRAR_MENSAJE_VICTORIA
    MOSTRAR_PALABRA_SECRETA
SINO (INTENTOS_RESTANTES == 0):
    MOSTRAR_MENSAJE_DERROTA
    MOSTRAR_PALABRA_SECRETA
FIN SI
```

FIN

7.- Referencias

- Universidad Veracruzana (mayo 2020). *Diagrama de Flujo*. UV. <https://www.uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf>
- Cintia Perazo (23 de mayo de 2024). *Ocho estrategias para ganar siempre al ahorcado*. La Nación. <https://www.lanacion.com.ar/juegos/trucos-y-consejos/ocho-estrategias-para-ganar-siempre-al-ahorcado-nid23052024/>
- Thecodest. *Capa Lógica de Negocio: Una definición exhaustiva*. THE CODEST. <https://thecodest.co/es/diccionario/capa-de-logica-empresarial/>