Proyecto 3 – Paradigma Lógico: Ahorcado en Prolog

Nombre del Proyecto: Ahorcado en Prolog

Curso: Lenguajes de Programación

Estudiantes:

Alberto Alvarez Badilla - 2023071564 Alfredo Mercado Rios - 2023377984

Profesor: Bryan Tomas Hernandez Sibaja

Fecha de entrega: 13 de junio de 2025

Institución: Instituto Tecnológico de Costa Rica (TEC)

Índice

Enlace de GitHub

A continuacion, se presenta el enlace del repositorio con el proyecto:

https://github.com/BetilloDelProyecto/Proyecto3Lenguajes.git

Pasos de instalación del programa

1. Instalar SWI-Prolog
2. Clonar el repositorio desde GitHub:

```
git clone https://github.com/BetilloDelProyecto/Proyecto3Lenguajes.git
cd ahorcado-prolog
```

3. Asegurarse de tener un archivo `palabras.txt` en la misma carpeta, con una palabra por línea.
4. Ejecutar el programa desde consola:

```
swipl -q -f main.pl -t halt.
```

Manual de usuario

Menú Principal:

Al ejecutar el programa, se mostrará el siguiente menú:

```
=== Juego Ahorcado ===
1. Empezar a Jugar
2. Configuracion
3. Salir
```

Opción 1 – Empezar a Jugar:

- El sistema selecciona una palabra secreta al azar de la base de conocimiento.
- El usuario debe adivinar letras, una por una, con un número limitado de intentos.
- Se gana al descubrir la palabra completa, o se pierde si se acaban los intentos.

Opción 2 – Configuración:

- **Agregar Palabra:** Permite añadir nuevas palabras a la base (`palabras.txt`) si no existen aún.
- **Número de Intentos:** Permite configurar la cantidad de intentos permitidos antes de perder.

Opción 3 – Salir:

- Termina la ejecución del programa.

Arquitectura lógica utilizada

Estructura general

El sistema está dividido en **3 archivos principales**:

Archivo	Función principal
`main.pl`	Punto de entrada. Contiene los menús y control.
`palabras.pl`	Módulo de persistencia (cargar/guardar palabras).
`logica.pl`	Toda la lógica del juego.

Módulo `palabras.pl`

Encargado de **gestionar la base de conocimiento** (`palabras.txt`). Incluye:

- `cargar_palabras/2`: Carga las palabras desde un archivo de texto a una lista.
- `guardar_palabras/2`: Guarda una lista de palabras al archivo.
- `agregar_si_no_existe/3`: Agrega una palabra a una lista si no está repetida.

Este módulo asegura persistencia en disco y evita duplicados.

Módulo `logica.pl`

Contiene toda la **lógica del juego Ahorcado**, sin ciclos imperativos:

- `seleccionar_palabra_aleatoria/1`: Elige una palabra al azar desde la base.
- `estado_inicial/2` y `estado_inicial/3`: Crea el estado inicial del juego.
- `mostrar_palabra/3`: Muestra la palabra con guiones y letras descubiertas.
- `adivinar_letra/4`: Actualiza el estado según si la letra es correcta, incorrecta o repetida.
- `verificar_estado/2`: Verifica si se ganó, se perdió, o el juego continúa.
- `jugar/1`: Bucle recursivo que representa un turno de juego.
- `siguiente_turno/1`: Llama recursivamente a `jugar/1` o finaliza el juego.
- `leer_letra/1`: Lee y valida entrada del usuario (una sola letra).

Archivo principal `main.pl`

- Muestra el menú principal y llama a las funciones correspondientes de lógica o configuración.
- Usa `dynamic/1` para permitir cambiar el número de intentos en tiempo de ejecución.
- Se asegura de que el juego se pueda reiniciar y configurarse fácilmente desde consola.

Flujo del juego

1. El usuario escoge “Empezar a jugar”.
2. Se selecciona una palabra secreta de forma aleatoria.
3. Se inicializa el estado del juego (`estado/3`).
4. Se entra en un ciclo de turnos (`jugar/1`) donde:
    - Se muestra el progreso de la palabra.
    - El usuario introduce una letra.
    - Se actualiza el estado del juego.
    - Se verifica si hay victoria, derrota o si se continúa.
5. Al final, se vuelve al menú principal.
