Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Lenguajes de Programación - IC4700
I Semestre de 2025

Estudiantes:

- Sebastián Calvo Hernández - 2022099320
- Isaac Gamboa Ureña - 2022437592

Profesor:

- Bryan Hernández Sibaja

Índice

Juego de Ahorcado (Programación Lógica)

Este proyecto implementa el clásico juego de Ahorcado en Prolog. El objetivo es adivinar una palabra secreta seleccionando letras, con un número limitado de intentos. El juego incluye dos modos: uno con una palabra aleatoria y otro donde el jugador ingresa su propia palabra secreta.

Repositorio del proyecto en GitHub: https://github.com/SebasCH04/hangman-prolog

---

Pasos de instalación del programa

Requisitos previos

1. **Instalar SWI-Prolog**:
   Asegúrate de tener **SWI-Prolog** instalado en tu sistema. Puedes descargarlo desde el sitio web oficial: https://www.swi-prolog.org/Download.html.

2. **Sistema operativo**:

   - Windows, Linux o macOS son compatibles.

Pasos para instalar

1. **Clona o descarga el repositorio**:
   - Si usas Git, clona el repositorio con:

     ```
     git clone https://github.com/SebasCH04/hangman-prolog
     ```

   - O si prefieres, puedes descargar el proyecto como un archivo ZIP.
2. **Abrir el archivo en SWI-Prolog**:
   - Navega al directorio donde descargaste el proyecto en tu terminal o abre el archivo en el entorno de desarrollo.
   - Inicia **SWI-Prolog** e ingresa:

     ```
     ?- [palabras], [display], [game_logic], [main].
     ```

3. **Ejecutar el juego**:
   - Para iniciar el juego, ingresa:

     ```
     ?- run.
     ```

Manual de usuario

Descripción del juego

El objetivo del juego es adivinar una palabra secreta letra por letra. El jugador tiene un número limitado de intentos para hacerlo. Por cada letra correcta, se muestra en la palabra, mientras que por cada intento incorrecto, el número de intentos restantes disminuye.

Pasos para jugar:

1. **Selecciona el modo de juego**:
   - **Modo 1**: Palabra aleatoria (el juego elige una palabra al azar de la lista de palabras).
   - **Modo 2**: Ingresar palabra secreta (el jugador introduce su propia palabra secreta).
2. **Adivina las letras**:
   - El jugador debe ingresar una letra en cada turno.
   - El juego muestra las letras adivinadas correctamente y la cantidad de intentos restantes.
3. **Fin del juego**:

- El juego termina cuando el jugador adivina todas las letras de la palabra secreta o se queda sin intentos.

Arquitectura lógica utilizada

Explicación del funcionamiento

El programa está diseñado utilizando Prolog, lenguaje de programación lógica, que permite realizar la implementación de un juego como el "Ahorcado" de manera sencilla. La lógica del juego se organiza en varios módulos que interactúan entre sí. A continuación, se detalla la arquitectura y el funcionamiento.

1. **Módulos principales**:
   - `palabras.pl`:
     Este módulo contiene la lista de palabras que el juego utiliza. Puede seleccionar palabras aleatorias o permitir que el jugador ingrese una palabra secreta.
   - `display.pl`:
     Este módulo es responsable de mostrar el estado actual del juego, como las letras adivinadas y el número de intentos restantes.
   - `game_logic.pl`:
     Contiene la lógica principal del juego. Aquí se maneja el flujo del juego, como la selección de la palabra secreta, el número de intentos, la validación de las letras adivinadas y la finalización del juego.
   - `main.pl`:
     Es el punto de entrada del programa. Permite iniciar el juego, leer las opciones del jugador y llamar a los predicados correspondientes para gestionar el juego.
2. **Flujo de ejecución**:
   - El programa comienza con el predicado `run/0` en `main.pl` que muestra un menú para que el jugador elija el modo de juego.
   - Luego, el predicado `start/0` en `game_logic.pl` inicia el juego, eligiendo la palabra y comenzando el ciclo de adivinanza.
   - A medida que el jugador adivina letras, el estado del juego se actualiza utilizando los predicados de `display.pl` y `game_logic.pl`.
   - El juego termina cuando el jugador adivina la palabra correctamente o se queda sin intentos.

Estructura lógica del juego:

- `start/0`: Inicializa el juego, selecciona la palabra y empieza el ciclo de juego.
- `play/3`: El ciclo principal del juego donde se verifica si el jugador ha ganado o perdido y se llama a `process_guess/5` para manejar cada intento.
- `process_guess/5`: Procesa cada intento del jugador, valida si la letra adivinada es correcta o incorrecta y actualiza el número de intentos restantes.