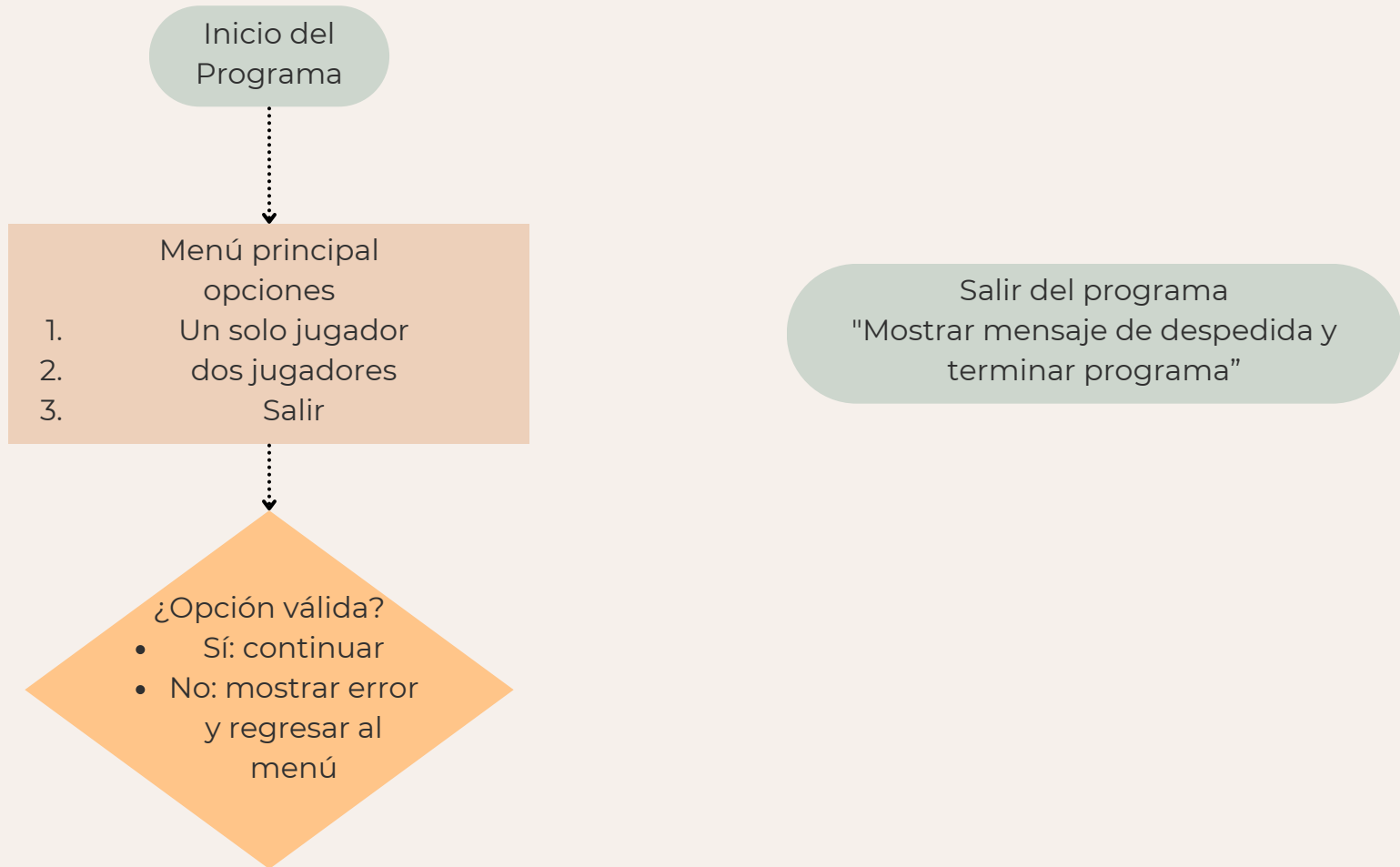
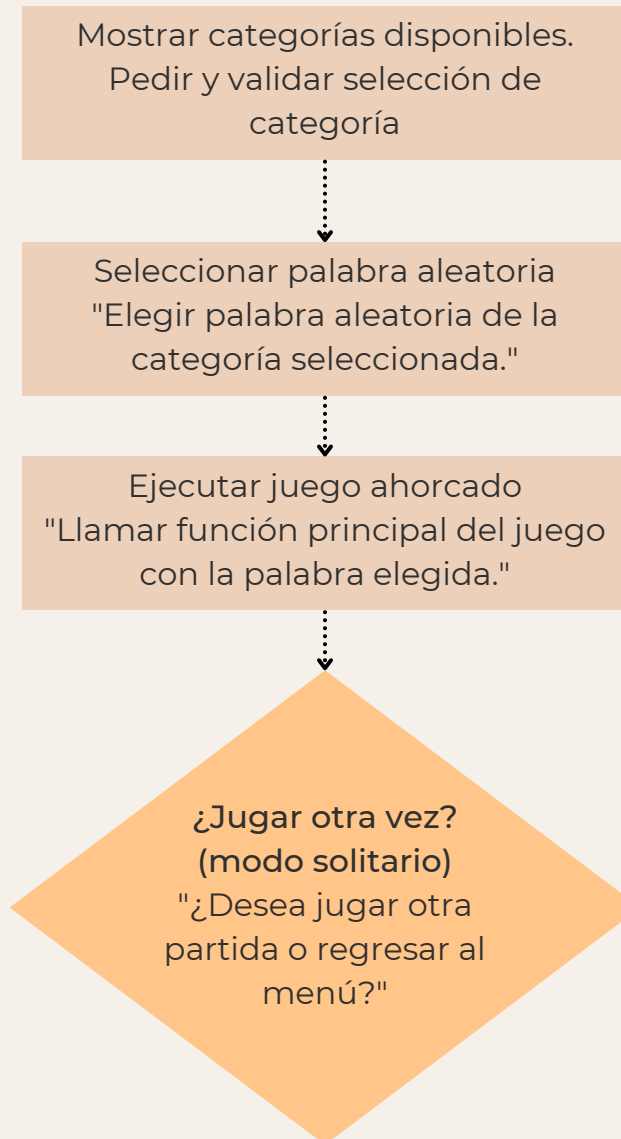


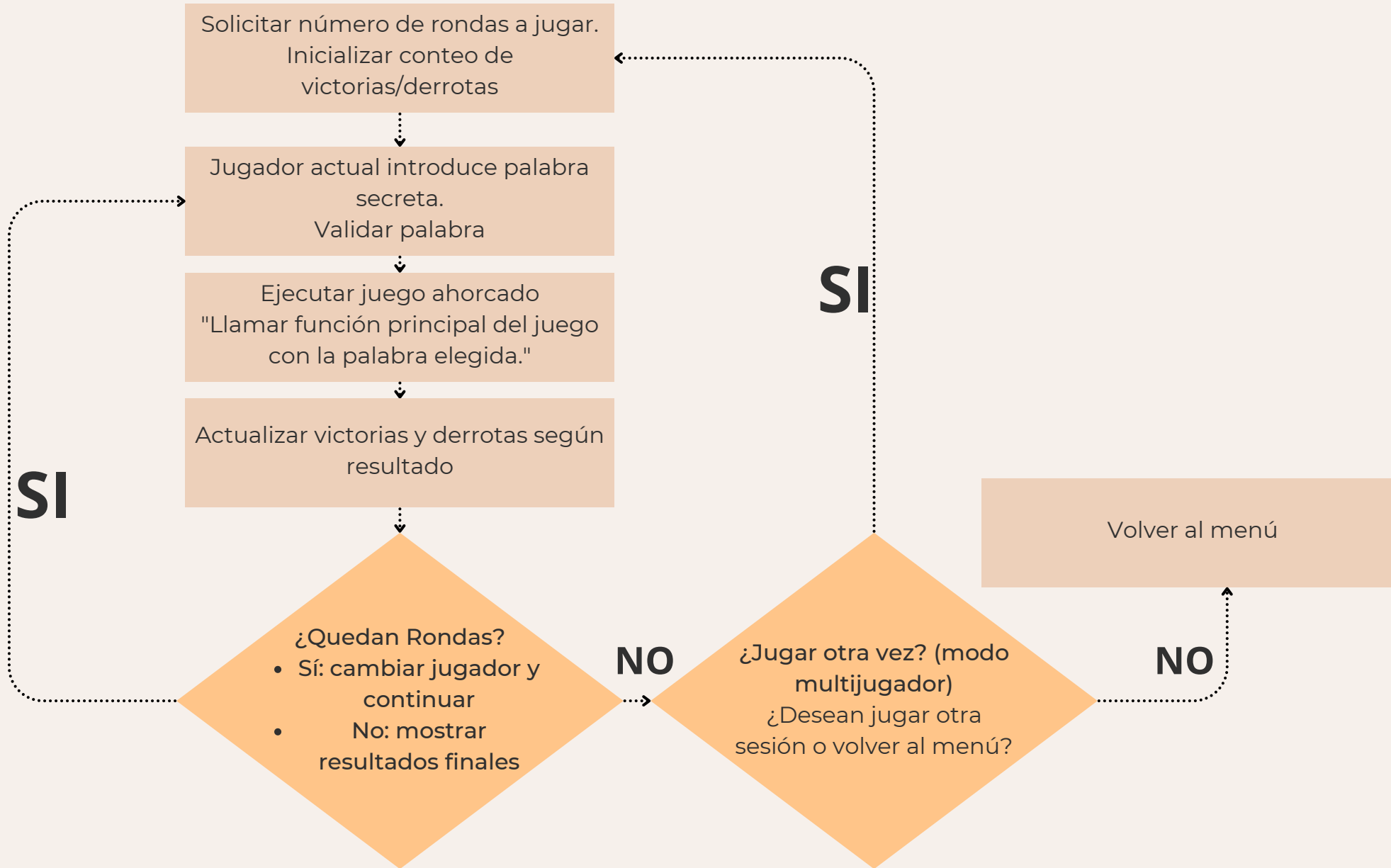
# Inicio del programa y muestra menú principal



# Modo Solitario



# Modo Multijugador



# Conclusiones

- El código implementa un juego del ahorcado funcional y amigable, con opciones para jugar solo y en modo multijugador, lo que amplía su alcance y dinamismo para distintos usuarios.
- Se aplicó una buena estructuración modular mediante funciones específicas que cumplen tareas puntuales, facilitando la lectura, mantenimiento y posible ampliación del programa.
- La constante validación de entradas evita errores y garantiza que el juego se desarrolle sin interrupciones inesperadas, mejorando la experiencia del jugador.
- La selección aleatoria de palabras dentro de categorías brinda variedad e interés en cada partida, manteniendo la dinámica fresca y entretenida.
- La implementación del control de intentos y el manejo de letras adivinadas simula fielmente el juego tradicional, adaptado al entorno digital mediante programación.
- La elaboración previa del diagrama de flujo fue fundamental para entender y planificar la lógica del juego paso a paso, optimizando así la codificación y asegurando un flujo claro y eficiente.
- La estructura del código permite agregar fácilmente nuevas funcionalidades, como más categorías, palabras, o ajustes en la dificultad, sin tener que rehacer la base del programa.

# Limitaciones e Implicaciones del Proyecto

## Limitaciones del Proyecto

- El juego tiene un conjunto limitado de palabras y categorías, lo que puede hacer que la experiencia sea repetitiva a largo plazo.
- La interfaz es simple y basada en consola, sin gráficos ni animaciones que mejoren la interacción visual.
- Los parámetros del juego, como el número de intentos, son fijos y no hay opción para personalizar la dificultad.
- Se valida la entrada del usuario de forma básica, pero no se manejan casos muy complejos o posibles fallos en inputs inusuales.
- No existe almacenamiento persistente para guardar resultados o progreso entre sesiones.

## Implicaciones del Proyecto

- Permite aprender y aplicar conceptos básicos de programación como funciones, ciclos, y manejo de datos.
- Introduce la lógica de juegos y control de flujo, fundamentales para futuros proyectos más complejos.
- Refuerza la importancia de validar entradas para evitar errores y mejorar la experiencia del usuario.
- Establece una base sólida para agregar funcionalidades futuras, como interfaces gráficas o más opciones de juego.
- Desarrolla habilidades de planificación con diagramas de flujo, mejorando la organización y claridad en la programación.