

# INGENIERÍA DE APLICACIONES 2025

## PROYECTO - HOME SAFETY HOTLINE

Fecha de presentación preliminar: **21 de Noviembre**

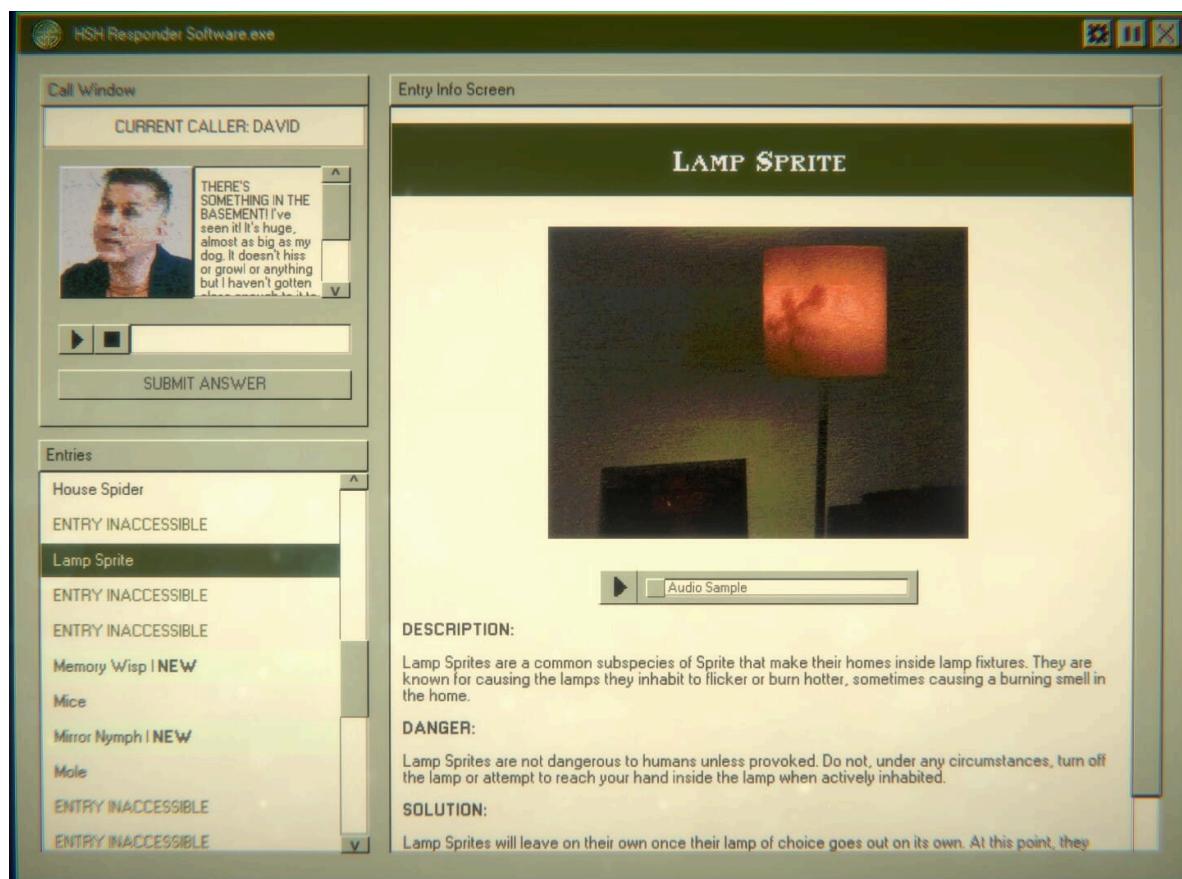
FECHA DE ENTREGA Y EXPOSICIÓN: **28 de Noviembre**

### EL PROYECTO PODRÁ REALIZARSE EN COMISIONES DE 1 O 2 ALUMNOS.

El presente proyecto tiene como objetivo diseñar y desarrollar un juego estilo "Home Safety Hotline" ([https://store.steampowered.com/app/2357910/Home\\_Safety\\_Hotline/](https://store.steampowered.com/app/2357910/Home_Safety_Hotline/)). En este juego, el jugador asume el rol de operador de una línea telefónica de una empresa de control de plagas. Diversos clientes llaman para reportar síntomas relacionados con la presencia de una plaga (sin mencionar específicamente cuál es). A partir de los síntomas descritos y la información disponible sobre cada tipo de plaga, el jugador debe identificar la plaga responsable y proporcionar instrucciones precisas sobre cómo solucionar el problema.

La pantalla principal del sistema se compone de tres secciones:

1. Una ventana que muestra las llamadas en espera.
2. Una ventana para seleccionar las diferentes entradas del sistema, que en este caso representan distintas plagas.
3. Una ventana que presenta la información detallada de la entrada o plaga seleccionada en la sección anterior.



Periódicamente, una nueva persona llama para reportar un problema. El jugador debe poner a la persona en espera para investigar la información disponible sobre las plagas y deducir cuál está causando el inconveniente. Una vez que el jugador ha identificado la plaga, debe comunicar su diagnóstico a la persona que llama, finalizando así la llamada. Más adelante, esa misma persona volverá a contactar para informar si el consejo proporcionado fue efectivo o no.

## GAMEPLAY

([https://www.youtube.com/watch?v=4\\_DZfnRjPbw&t=818s&pp=ygUNY2FkZSBob3QgbGluZQ%3D%3D](https://www.youtube.com/watch?v=4_DZfnRjPbw&t=818s&pp=ygUNY2FkZSBob3QgbGluZQ%3D%3D)).

## DÍAS

El juego consistirá de 3 días:

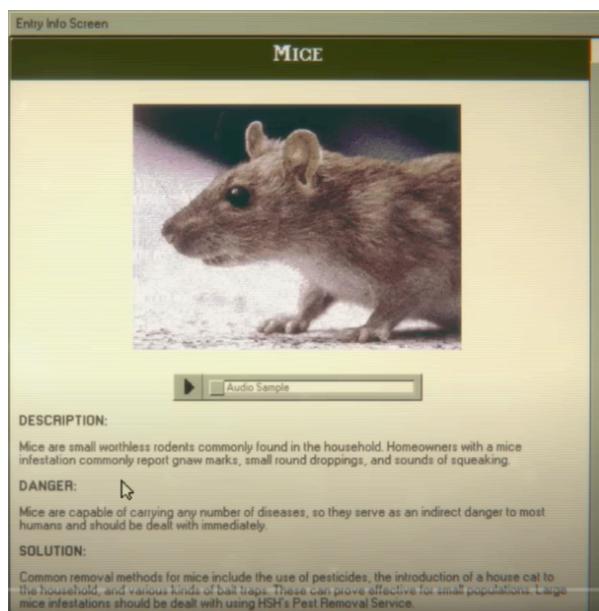
- **Primer día:** sólo habrá 5 plagas disponibles de tipo Normal (ver más abajo). Llamarán 7 personas solicitando ayuda.
- **Segundo día:** se agregarán a la lista de plagas 5 plagas más de tipo Extraño. Llamarán 8 personas este día solicitando ayuda.
- **Tercer día:** se agregarán a la lista de plagas 5 plagas más de tipo Especial. Llamarán 10 personas este día solicitando ayuda.

Es importante mencionar que las llamadas de cada día podrán reportar cualquier tipo de plaga disponible, no necesariamente las que se agregaron ese mismo día. Es decir, por ejemplo, las personas que llamen el día 3 podrán reportar plagas tanto del día 1, 2 o 3.

## PLAGAS

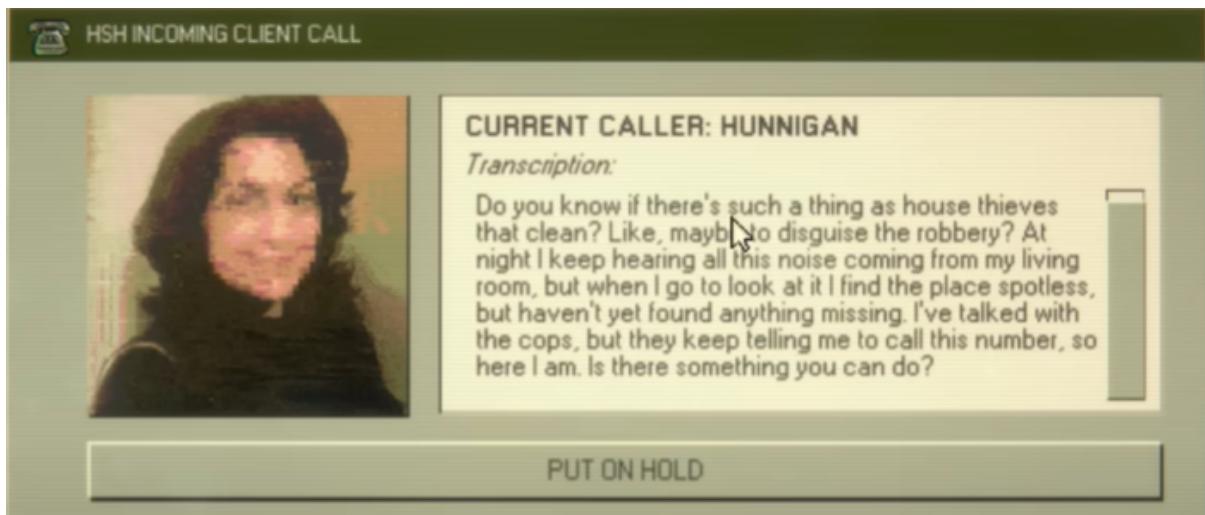
Cada plaga en el juego contará con un **nombre**, una **imagen**, una **descripción**, **peligro** y una **solución** específica. Habrá tres tipos de plagas: Normales, Extrañas y Especiales.

- **Normales:** La primera tanda de 5 plagas estará basada en problemas cotidianos (como moho, hongos, moscas, entre otros, según la temática seleccionada).
- **Extrañas:** La segunda tanda de 5 plagas incluirá elementos un poco más inusuales pero "posiblemente reales" (como auroras hogareñas, duendes topos, etc.).
- **Especiales:** La tercera tanda de 5 plagas se centrará en fenómenos mucho más extraños o paranormales.



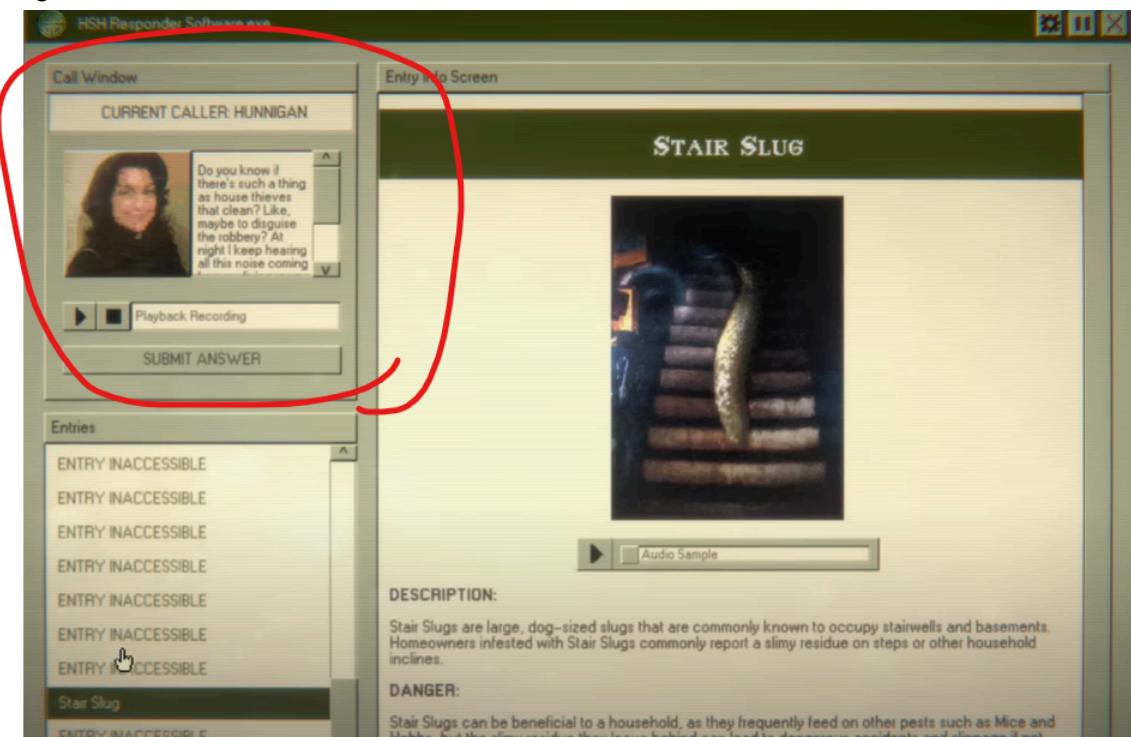
## LLAMADAS

Cada llamada dispondrá del **nombre** y la **imagen** de la persona que llama, así como también el **mensaje** correspondiente.



Existen 3 tipos de llamadas.

1. **Llamadas de personas pidiendo consejos:** Estas llamadas quedarán en espera hasta que se les proporcione una respuesta. Si hay más de una llamada en espera, se organizarán en una cola, de modo que se respondan en el orden en que llegaron (ver imagen)



2. **Llamadas de confirmación:** Estas llamadas provienen de personas que previamente recibieron un consejo, informando si la ayuda proporcionada fue efectiva o no.
3. **Llamadas extras:** Estas son llamadas que no buscan asistencia, sino que tienen el propósito de informar sobre algún hecho o misterio relacionado con el juego. Estas llamadas se cierran y listo.



## INFORMACIÓN

En cuanto a la gestión de los datos del juego, se deberá elegir entre dos opciones:

- **Opción 1: Información Estática**

En esta opción, toda la información del juego (como plagas, personas, llamadas, imágenes, etc.) se almacenará en archivos almacenados online, que el juego descargará al iniciarse. Esto requerirá la creación de tres versiones diferentes del juego, cada una con una temática distinta. La información estática sobre el juego podrá ser creada utilizando uno o más LLMs, como por ejemplo ChatGPT.

- **Opción 2: Información Dinámica**

En esta alternativa, toda la información del juego (plagas, personas, llamadas, imágenes, etc.) se generará dinámicamente utilizando uno o más LLM, como por ejemplo ChatGPT.

En este caso, cada vez que se juegue, toda la información presentada será nueva, haciendo que la experiencia de juego sea diferente cada vez.

En ambos casos, la aplicación compilada no contendrá nada de la información del juego. Se deberá detallar en la presentación la forma en que se obtuvo toda la información del juego.

## OTROS REQUERIMIENTOS DE IMPLEMENTACIÓN

- Todo el texto asociado del juego debe estar en español. La aplicación deberá permitir seleccionar al menos 2 idiomas adicionales, haciendo la **traducción correspondiente en tiempo real**.
- Implementar un sistema de **Text to Speech** que lea en voz alta cada diálogo de los personajes. Esta característica debe poder activarse y desactivarse en cualquier momento.
- El sistema deberá tener un **reloj digital** que muestre la hora actual del sistema.
- La aplicación deberá exportarse a un ejecutable .exe para correr en windows. En caso de que no se pueda, la aplicación deberá funcionar en la web (se puede utilizar play.unity.com, githubPages, etc.).

## OBJETIVOS DE DISEÑO

- Implementar el **patrón de arquitectura Modelo - Vista - Presentador (MVP)**.
- Cumplir con los principios **SOLID** y de **Clean Code**.
- Organizar el proyecto en **paquetes** adecuados (namespaces en C#).
- Presentar un **diagrama de clases UML** y otro **diagrama de clases simplificado**.

## MODALIDAD DE ENTREGAS

La presentación preliminar consiste en que cada comisión muestre sus avances hasta el momento. Para dicha fecha se recomienda tener todo el diseño terminado, un gran avance en cuanto a la implementación del sistema, y un borrador de la presentación oral.

En el día pautado se deberá realizar una exposición oral que debe incluir las siguientes secciones (el orden puede variar):

- a. Definición y objetivos de la actividad.
- d. Diseño del sistema (decisiones de diseño, patrones aplicados, etc.).
- e. Implementación (detalle de las tecnologías utilizadas).
- f. Desafíos presentados.
- g. Alcance de su solución (incluye muestra funcional del sistema).
- h. Conclusiones (logros alcanzados, tareas por realizar, pasos a seguir, etc.).

Además, **debe asegurarse** de subir al repositorio de github asignado los siguientes archivos:

1. La presentación de diapositivas de la exposición oral.
2. El código fuente de la aplicación.
3. Diagramas realizados.

## VIDEO

Se deberá crear y subir al repositorio un pequeño video (de no más de 1 minuto) a modo de Trailer del juego, o similar.

## EVALUACIÓN

La actividad será calificada evaluando:

- La participación en las actividades relacionadas que se plantean en la práctica.
- La presentación oral.
- El avance en la presentación intermedia.
- El diseño, la calidad y la prolijidad del mismo.
- La calidad del diseño y del código.
- La interfaz de usuario y la usabilidad de la misma.

Los commits realizados en el repositorio GitHub representan un historial de cambios y de avance del proyecto. Dichos commits deberán ser representativos a los cambios efectuados.

## CRONOGRAMA TENTATIVO

PARA LA FECHA	SE RECOMIENDA HABER REALIZADO
21/10	<p>Planificar la temática del juego. Ir generando el texto y las imágenes correspondientes si se va a realizar la versión local.</p> <p>Explorar cómo generar texto o imágenes de manera dinámica con LLMs si se va a utilizar esa opción.</p>
24/10	Realizar un diagrama básico del sistema pensando en todos los distintos componentes del mismo. Consultar con la cátedra.
18/10	<p>Realizar un mockup de las pantallas de la aplicación, es decir, ir pensando en las distintas pantallas y en cómo se van a comunicar. Consultar el mockup con la cátedra.</p> <p>Ir investigando cómo implementar el reloj, la traducción y el text-to-speech.</p>
31/10	Completar el diseño del sistema con la parte del MVP, pensando en la comunicación entre las distintas vistas y el modelo.
04/11	<p>Implementar las interfaces gráficas de la aplicación y las conexiones entre los botones y las pantallas.</p> <p>Crear un ejecutable y probar que funcione.</p>
07/11	<p>Seguir mejorando el diseño del sistema.</p> <p>Seguir implementando el resto de la lógica del juego.</p> <p>Ir probando cada tanto el poder crear un ejecutable y que siga funcionando.</p>
11/11	Implementar un nivel del juego que funcione.
14/11	Comenzar a preparar la presentación.
17/11	Incorporar las distintas partes del juego en la aplicación final.
21/11	Asegurar que el código cumpla con los objetivos del proyecto. Pulir y refactorizar el código en caso de ser necesario.
24/11	Terminar de completar y pulir los diagramas.
28/11	<p>Crear el ejecutable de la aplicación.</p> <p>Crear el video.</p> <p>Terminar de pulir la presentación.</p>