

Capítulo 1

Desarrollo de la aplicación del test subjetivo.

En este capítulo se explican las herramientas y lenguajes utilizados para la creación de la aplicación de escritorio con la que se realizan posteriormente los test subjetivos de audio necesarios para el proyecto. Así mismo, se explica el algoritmo utilizado y las diferentes versiones utilizadas para la creación de la interfaz de usuario.

1.1. Lenguajes y programas utilizados

Para el desarrollo de la aplicación, se ha optado por utilizar el lenguaje de programación Python3. Este lenguaje tiene la ventaja de que incluye numerosas librerías que permiten trabajar con interfaces gráficas con las que los participantes de los test puedan interactuar de forma sencilla. Para nuestro caso particular, se ha optado por utilizar la librería del entorno gráfico GTK (la librería se llama *PyGObject*¹) que es una de las más utilizadas en entornos Linux, aunque también puede utilizarse en entornos de Windows o Mac.

Para el diseño de la interfaz, se podría haber codificado directamente en lenguaje python. No obstante, se ha utilizado el programa *Glade*² que permite utilizar un entorno gráfico para la creación de todos los elementos de la interfaz. Con dicho programa se obtiene un fichero XML con extensión “.glade” que es el que el *script* de Python lee y con el que se genera la interfaz que el usuario utiliza. En la figura 1.1 se puede observar la pantalla de inicio de dicho programa.

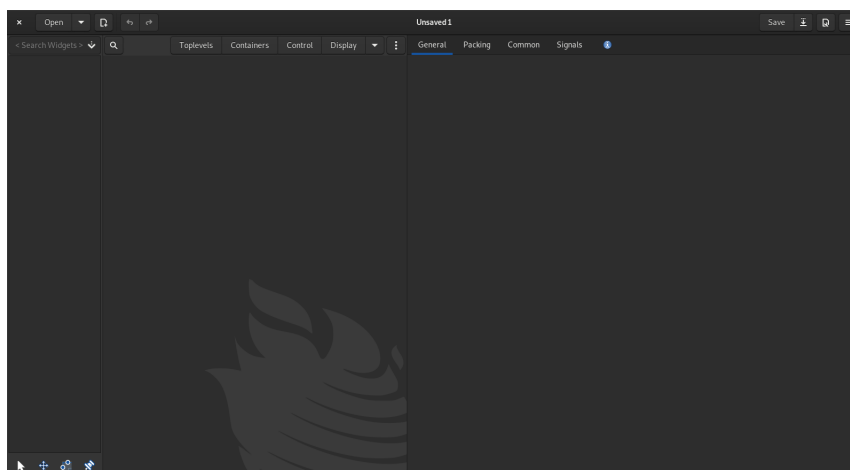


Figura 1.1: Ventana de inicio del programa *Glade*.

¹<https://pygobject.readthedocs.io/en/latest/>

²<https://glade.gnome.org/>

Para la codificación de los scripts de Python se ha utilizado el programa *Gnome Builder*³; un entorno de aplicaciones que incluye todas las herramientas para depurar, compilar y ejecutar los scripts dentro del mismo espacio. Al igual que con *Glade*, el programa es gratuito de código abierto. En la figura 1.2 se encuentra una captura de la interfaz del programa.

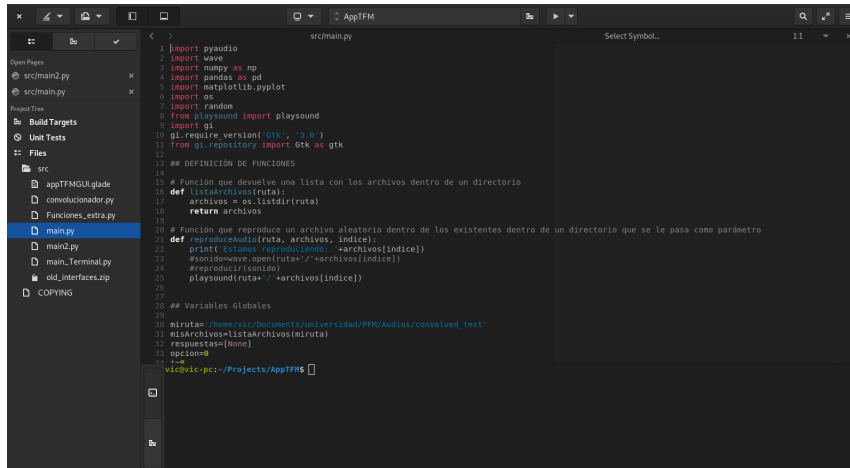


Figura 1.2: Captura del programa *Gnome Builder*.

1.2. Características de la interfaz de usuario

Como se ha comentado en el apartado anterior, la interfaz de usuario se ha diseñado utilizando el programa *Glade*. Desde un principio, se ha querido diseñar una interfaz que sea lo más simple posible, no sólo por facilidad para realizarla, sino para que las personas participantes puedan centrarse exclusivamente en los objetivos del test y hacer que su uso no suponga ninguna dificultad.

En una primera instancia se decidió generar una interfaz en la que los elementos principales fueran dos botones con los que el usuario pudiera escoger cuál de los audios quería reproducir. Las respuestas se recogerían en dos casillas de verificación, o *toggles*. También se incluyó un botón en la parte inferior para enviar cada una de las respuestas. En la parte superior se incluye un texto que indica el número de pregunta por la que va el test. Todos estos elementos se pueden observar en la figura 1.3.

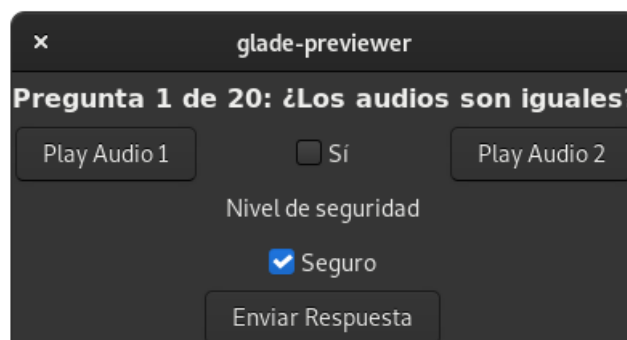


Figura 1.3: Versión inicial de la interfaz de usuario.

Esta configuración es la que se utiliza en el test previo. Al finalizar dicho test, se les pidió a los participantes que propusieran diferentes mejoras para perfeccionar la interfaz de usuario. Entre ellas, las más habituales consistieron en la sustitución de los *toggles* por elementos más grandes al estilo de *switches* o interruptores y la actualización de la posición de dichos elementos a su estado inicial entre

³<https://apps.gnome.org/app/org.gnome.Builder/>

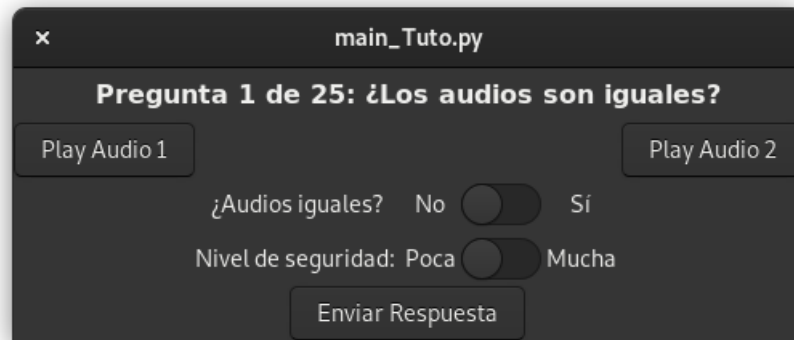


Figura 1.4: Versión final de la interfaz de usuario.

cada una de las preguntas. Atendiendo a estas propuestas, se modifica la interfaz con el resultado que se muestra en la figura 1.4.

1.3. Codificación del script en Python

El procedimiento seguido para la codificación de la aplicación ha sido gradual. En primer lugar se han diseñado diferentes métodos y funciones que se apliquen directamente desde la consola. De esta forma, se comprueba que el algoritmo funciona correctamente y que se extraen los resultados en el formato deseado.

Una vez, se ha comprobado que este *script* funciona, se modifica el mismo haciendo que las funciones sean llamadas por los diferentes elementos de la interfaz de usuario ya creada y que se referencian dentro del código de Python.

Durante todo el proceso, se realizaron consultas a la documentación de la API de Python para GTK⁴ y de las diferentes librerías necesarias para el desarrollo de la aplicación. Varios ejemplos de estas librerías son:

- **numpy**⁵: para el trabajo con matrices al estilo de programas como *Matlab*.
- **pandas**⁶: para la edición y creación de estructuras de datos.
- **os**⁷: para la navegación dentro de sistemas de ficheros del sistema operativo.
- **gi**⁸: para el manejo de los elementos gráficos y la gestión de sus señales.

Por último durante el periodo de tiempo entre los dos tests subjetivos de audio se añadió la funcionalidad que permitía reproducir los dos audios pulsando las teclas “1” y “2” del teclado respectivamente sin necesidad de utilizar el ratón. Esta funcionalidad se incorporó como sugerencia por parte de varias de las personas participantes que comunicaron las dificultades de escuchar los audios con los ojos cerrados (decisión personal) al tener que pinchar los botones con el ratón del ordenador.

1.3.1. Estructura del código

El código de Python se encuentra dividido en diferentes partes: En primer lugar, la zona donde se importan las diferentes librerías para el correcto funcionamiento del script. Aquí se incluyen las que

⁴<https://python-gtk-3-tutorial.readthedocs.io/en/latest/>

⁵<https://numpy.org/>

⁶<https://pandas.pydata.org/>

⁷<https://docs.python.org/3/library/os.html>

⁸<https://pygobject.readthedocs.io/en/latest/guide/api/api.html>

permiten reproducir los diferentes audios, manejar información, navegar por el sistema de archivos del ordenador, la interfaz gráfica, etc.

A continuación, se encuentra una zona donde se definen diferentes funciones que serán usadas repetidamente a lo largo del *script*.

Después, se definen algunas variables globales necesarias y da comienzo la clase *Main* de la aplicación. En ella, se inicializan las referencias a los diferentes elementos de la interfaz gráfica con las que los usuarios interactúan y se establecen qué funciones se llaman en función de las señales que manejan. Estas funciones se definen en último lugar a continuación de dichas inicializaciones. En el anexo C se puede leer el código en su totalidad.

Bibliografía