

# **Seminario 1 DDSI: Acceso a bases de datos en el desarrollo de Sistemas de Información.**

## **Aplicación accesible a base de datos mediante interfaz**

**Autores: Requisitos Exquisitos.**

***Julián Garrido Arana***

***Juan Valentín Guerrero Cano***

***Joaquín Arcila Pérez***

***Gerardo Arenas Nasrawin***

***Daniel Cerezo García***

Para realizar la aplicación hemos elegido como lenguaje de programación Python, por su versatilidad y el gran número de librerías y ejemplos que ofrece.

A la hora de implementar la aplicación hemos creado inicialmente la interfaz de usuario, para poder visualizar las acciones especificadas como tarea y a partir de esta implementar el backend pudiendo debugear la app de forma más intuitiva y visual.

Para programar la interfaz hemos hecho uso de la biblioteca Tkinter de Python, por lo que, previamente, tuvimos que [instalarla](#) en el dispositivo. Al ser la primera vez que hacíamos uso de este paquete, buscamos un [tutorial](#) de inicialización sobre el mismo. Además, nos informamos sobre la destrucción de ventanas, para tener solo una de ellas activa ([web](#)), posicionamiento de widgets ([web](#)) y sobre la admisión de usuarios ([web](#)).

Por otro lado, para programar la parte “backend” de la aplicación hemos creado un archivo aparte que se encarga de ejecutar todas las sentencias SQL, para llevar a cabo las funcionalidades que la interfaz ofrece (y que se requieren en el seminario).

Sin embargo, antes de escribir código, estudiamos la arquitectura que debería tener nuestro sistemas, es decir; qué Driver Manager debíamos utilizar y qué Driver debíamos instalarnos para poder hacer uso correcto de la BD. Como el SGBD era Oracle e íbamos a trabajar con python en linux, escogimos ODBC (unixODBC más concretamente) como Driver Manager. Como conexión entre el Driver Manager y el SGBD (Driver) investigamos entre los muchos que ofrecía Oracle, decantándose por el que parecía más comúnmente utilizado y más fácil de instalar, “Devart ODBC Driver for Oracle”. Para la instalación, seguimos los comandos descritos en la propia [documentación](#) del Driver, que se tratan, básicamente, de descargar el paquete y ejecutar, desde la terminal, la instalación del mismo.

A continuación, una vez instalado el Driver que hiciese de enlace entre el Driver Manager y el SGBD, estudiamos las sentencias de conexión proporcionadas por la librería de Python para ODBC (pyodbc). Gracias a la [documentación](#) de la librería y de los parámetros aportados en las diapositivas del seminario, conseguimos realizar una conexión a la BD mediante la función “connect”.

Finalmente, comenzamos con la parte de programación del BackEnd. Para ello, nos centramos en la arquitectura Vista-Controlador y, como el código correspondiente a la vista ya estaba realizado, nos centramos en crear un Controlador, el cual se encarga, mediante sentencias SQL (haciendo uso de la librería pyodbc), de llevar a cabo las funcionalidades que la interfaz ofrecía.

La clase “Controlador” nos permite modular la aplicación, pero, para su correcta implementación, nos fue necesario crear otra clase “AccesoBD”. Esta clase se encarga de realizar las sentencias explícitas SQL utilizando la librería pyodbc, y teniendo para ello una serie de métodos que nos permiten simplificar el acceso a la BD y separarlo de lo que es el controlador. Una vez creada esta clase pudimos utilizarla para conectarnos a la BD a través del constructor de la clase Controlador, y desde dicha conexión llevar a cabo las funcionalidades que la aplicación ofrece.

En cuanto al reparto de trabajo, Julián, Gerardo y Joaquín se han encargado de la implementación de la interfaz gráfica, mientras que Valentín y Daniel se han encargado de todo lo relacionado con la conexión a la BD y de la implementación del BackEnd. El documento del proyecto ha sido realizado bajo consenso de todos los miembros del equipo y con colaboración indistinguible de los cinco.