

# **Trabajo de Fin de Grado**



**UNIVERSIDAD  
DE GRANADA**

**Desarrollo de un Chatbot para  
Servicios Sociales Dirigido a Personas  
en Situación de Dependencia**

**Sprint 4**

**Claudia Salado Méndez**

# 1. Introducción

En este sprint ya se tiene creada la base de datos con todas las tablas necesarias para crear nuevas funcionalidades por ello en esta entrega el objetivo va a ser crear esas nuevas funcionalidades para la página web y el chatbot para que los usuarios puedan usar para más cosas el chatbot, no sólo para consultas.

## 2. Implementación

- **Primer Objetivo:**

El primer objetivo en este caso fue crear un menú en el lateral derecho de la web, este es del tipo hamburguesa, es decir las tres típicas rayitas horizontales que si se pulsa sale el menú desplegable con las distintas opciones.

La implementación de este fue sencilla se hizo en el html principal del chatbot, y no se tocó casi nada del css para hacerlo en un próximo sprint que será más centrado en la interfaz de usuario, la accesibilidad y seguridad.

En este menú aparecen los siguientes campos, añadir cita, añadir entrada al historial médico, añadir recordatorio y contacto, que se ha puesto por si en un futuro se quieren indicar datos del responsable pero actualmente no tiene ninguna funcionalidad ni hace ninguna actividad. Esta es la descripción de las opciones implementadas y cómo funcionan:

- **Añadir cita:**

Está lo que hace es añadir una nueva entrada a la tabla appointments, guardando el ID del usuario actual que está usando el chatbot, además al pulsar el botón llevará a los usuarios a un formulario que tendrán que rellenar con la fecha, en el que aparecerá un calendario y seleccionarán el día y la hora además de escribir el lugar de la consulta. Estos datos se guardan en la tabla.

Su funcionamiento: para lograr esto se ha tenido que crear una nueva función en la base de datos que lo que hace es hacer un INSERT con sql en la tabla appointment comprueba que los datos insertados corresponden con los campos que hay que rellenar y si la inserción es exitosa nos devuelve un mensaje de confirmación.

A parte en el código python del chatbot hay que crear dos funciones más, una que nos redirigirá al nuevo html correspondiente al formulario de las citas, y la otra crea la conexión con la base de datos, consigue el id del usuario actual y recoge los datos del formulario enviando todos estos a la función de la base de datos para que la inserte en la tabla.

En el nuevo html se crea el formulario correspondiente a las citas, que pide la fecha y el lugar en este caso, y realizará la solicitud Ajax al servidor flask para guardar los datos.

- **Añadir Recordatorio:**

Esta lo que hace es añadir una nueva entrada a la tabla reminders, guardando el ID del usuario actual que está usando el chatbot, además al pulsar el botón llevará a los usuarios a un formulario que tendrán que rellenar con la fecha, en el que aparecerá un calendario y seleccionarán el día y la hora además de escribir el nombre que le quieran poner al recordatorio y la frecuencia pensando en que si es tomarse un medicamento pues sea de la siguiente forma, nombre: Paracetamol, fecha: 04/07/2023 16:53 y frecuencia: cada 8 horas. Estos datos se guardan en la tabla.

Su funcionamiento: para lograr esto se ha tenido que crear una nueva función en la base de datos que lo que hace es hacer un INSERT con sql en la tabla reminders comprueba que los datos insertados corresponden con los campos que hay que rellenar y si la inserción es exitosa nos devuelve un mensaje de confirmación.

A parte en el código python del chatbot hay que crear dos funciones más, una que nos redirigirá al nuevo html correspondiente al formulario de los recordatorios, y la otra crea la conexión con la base de datos, consigue el id del usuario actual y recoge los datos del formulario enviando todos estos a la función de la base de datos para que la inserte en la tabla.

En el nuevo html se crea el formulario correspondiente a los reminders, que pide la fecha, el nombre y la frecuencia en este caso, y realizará la solicitud Ajax al servidor flask para guardar los datos.

- **Añadir Entrada al Historial Médico:**

Esta lo que hace es añadir una nueva entrada a la tabla history, guardando el ID del usuario actual que está usando el chatbot, además al pulsar el botón llevará a los usuarios a un formulario que tendrán que rellenar un cuadro de texto sobre la nueva actualización sobre su historial médico. Estos datos se guardan en la tabla.

Su funcionamiento: para lograr esto se ha tenido que crear una nueva función en la base de datos que lo que hace es hacer un INSERT con sql en la tabla history, comprueba que los datos insertados corresponden con los campos que hay que rellenar y si la inserción es exitosa nos devuelve un mensaje de confirmación.

A parte en el código python del chatbot hay que crear dos funciones más, una que nos redirigirá al nuevo html correspondiente al formulario del historial, y la otra crea la conexión con la base de datos, consigue el id del usuario actual y recoge los datos del formulario enviando todos estos a la función de la base de datos para que la inserte en la tabla.

En el nuevo html se crea el formulario correspondiente al historial, que pide el texto correspondiente a la nueva entrada en este caso, y realizará la solicitud Ajax al servidor flask para guardar los datos.

- **Segundo Objetivo:**

Ahora que los usuarios ya pueden ingresar sus próximas citas, algunos recordatorios incluso actualizar su historial médico, lo que se quiere conseguir ahora es cubrir la necesidad de estos de poder visualizar pues las citas que tienen pendientes o recordatorios que todavía están vigentes o ver su historial médico al completo.

Ya que estas funcionalidades se van a usar más frecuentemente se ha decidido que en vez de estar en el menú lateral de la página web se sitúan los tres botones debajo del panel de escritura de las preguntas para el chatbot y que al pulsar cada uno de estos botones se muestre por la conversación la información correspondiente al botón que se ha pulsado, si por ejemplo se ha pulsado “mis citas” pues que se muestre las que tienen una fecha posterior a la actual al igual con los recordatorios y cuando se pulse “mi historial médico” que muestre al usuario toda la información guardada en su historial médico por el chat.

Por lo tanto se han creado estas tres nuevas funcionalidades, Mis Citas, Mis Recordatorios y Mi Historial Médico. A continuación se va a explicar cada una de ellas y cómo se ha conseguido su funcionamiento. Actualmente la forma en la que se muestran es muy sencilla, en el siguiente sprint se profundizará más en ello para que sea más visual y vistoso.

- **Mis Citas:**

Lo que hace este botón es que al pulsarlo mostrará las citas en el chat del bot, da igual que ya se haya hablado con él aparecerá todo debajo de la conversación incluso si ya ha mostrado más información correspondiente a los otros botones. Mostrará todas las citas correspondientes al usuario que está usando el chatbot y cuya fecha sea mayor que la actual, es decir las citas anteriores a la fecha actual no aparecerán, sí estarán guardadas en la base de datos para tener la información por si hiciera falta en algún caso pero no aparecen al pulsar el botón.

Su funcionamiento es un poco más complejo, lo principal es crear la función de la base de datos que saca los datos que se requieren de la tabla appointment. Esta lo primero que hace es coger la hora local y guardarla en una variable, guardandola en el mismo formato que está la hora de la tabla citas para poder compararlas más tarde, después sacará los datos de la fecha y el lugar de la tabla appointment donde el usuario es el actual que está usando el chatbot y la fecha sea mayor a la actual es decir sean citas próximas no antiguas, esto devuelve en un map de citas.

En el fichero python del chatbot también se deberá incluir una función, en la que se conecte con la base de datos, se guarde el ID de la conversación se llame a la función que se había creado anteriormente para obtener las citas, con esto se recorrerá el map de citas creando el string para cada una de ellas con la información que se quiere que se muestre por pantalla este string es el que se le envía al html.

En el html se debe crear el nuevo botón para “Mis Citas” además de crear nuevo código JS en la parte de script del html para recibir el string que se ha creado con

anterioridad en el archivo Python y mandarlo como respuesta al pulsar el botón para que así se muestra en la conversación con el chatbot.

#### - **Mis Recordatorios:**

Lo que hace este botón es que al pulsarlo mostrará los recordatorios en el chat del bot, da igual que ya se haya hablado con él aparecerá todo debajo de la conversación incluso si ya ha mostrado más información correspondiente a los otros botones. Mostrará todos los recordatorios correspondientes al usuario que está usando el chatbot y cuya fecha sea mayor que la actual, es decir los recordatorios anteriores a la fecha actual no aparecerán, sí estarán guardados en la base de datos para tener la información por si hiciera falta en algún caso pero no aparecen al pulsar el botón.

Su funcionamiento es un poco más complejo, lo principal es crear la función de la base de datos que saca los datos que se requieren de la tabla reminders. Esta lo primero que hace es coger la hora local y guardarla en una variable, guardandola en el mismo formato que está la hora de la tabla reminders para poder compararlas más tarde, después sacará los datos de la fecha, el nombre y la frecuencia de la tabla reminders donde el usuario es el actual que está usando el chatbot y la fecha sea mayor a la actual es decir sean recordatorios próximos no antiguos, esto devuelve en un map de recordatorios.

En el fichero python del chatbot también se deberá incluir una función, en la que se conecte con la base de datos, se guarde el ID de la conversación se llame a la función que se había creado anteriormente para obtener los recordatorios, con esto se recorrerá el map de los recordatorios creando el string para cada uno de ellos con la información que se quiere que se muestre por pantalla este string es el que se le envía al html.

En el html se debe crear el nuevo botón para “Mis Recordatorios” además de crear nuevo código JS en la parte de script del html para recibir el string que se ha creado con anterioridad en el archivo Python y mandarlo como respuesta al pulsar el botón para que así se muestra en la conversación con el chatbot.

#### - **Mis Historial Médico:**

Lo que hace este botón es que al pulsarlo mostrará el historial médico en el chat del bot, da igual que ya se haya hablado con él aparecerá todo debajo de la conversación incluso si ya ha mostrado más información correspondiente a los otros botones. Mostrará todo el historial médico correspondiente al usuario que está usando el chatbot.

Su funcionamiento es un poco más complejo, lo principal es crear la función de la base de datos que saca los datos que se requieren de la tabla history. En esta lo que se hace es seleccionar las entradas de texto de la tabla history en las que el ID del usuario es igual al ID del usuario actual que está usando el chatbot, así devolverá un map del historial médico de dicho usuario

En el fichero python del chatbot también se deberá incluir una función, en la que se conecte con la base de datos, se guarde el ID de la conversación se llame a la función que se había creado anteriormente para obtener el historial médico, con esto se recorrerá el map del historial creando el string para cada una de las entradas con la información que se quiere que se muestre por pantalla este string es el que se le envía al html.

En el html se debe crear el nuevo botón para “Mi Historial Médico” además de crear nuevo código JS en la parte de script del html para recibir el string que se ha creado con anterioridad en el archivo Python y mandarlo como respuesta al pulsar el botón para que así se muestra en la conversación con el chatbot.

- **Pequeños Arreglos:**

Además de todo esto también se ha hecho una pequeña mejor al obtener el id del usuario para acceder a su conversación con el chatbot cuando se iniciaba sesión, ya que en la url se mostraba anteriormente el id de usuario y ahora se ha modificado para que sea más seguro y no se muestre.

### 3. Conclusión

En este sprint se han conseguido todos los objetivos que se esperaban, teniendo prácticamente toda la parte del código python y de la base de datos hecha. Pudiendo así enfocar los próximos sprint en la parte visual del html, para que la página web sea más sencilla y accesible para todo tipo de usuarios.

Todo esto se hará a través de los archivos css previamente creados en sprints anteriores, pero añadiendo mucha más complejidad para que no sea una página web tan básica.

Además habrá que comenzar a poner todas las preguntas y respuestas en el entrenamiento del JSON para ir modificando y especializando el chatbot a que si sea para un uso orientado a personas en situaciones de dependencia.