

Memoria técnica de la aplicación.

La aplicación ha sido desarrollada para el sistema operativo móvil Android y se ha usado el lenguaje de programación Java.

La interfaz gráfica de la app cuenta con 3 distintos menús en una barra de navegación situada en la parte inferior.

En el menú de home tenemos un acceso rápido a distinta información del museo y del universo de Juego de Tronos, como algunos de los personajes y familias más relevantes, algunos objetos e incluso materiales de los que están hechas armas que aparecen en la serie.

En el menú de qr, podremos acceder a un lector de QR que leerá los códigos qr que están en las distintas estatuas y objetos expuestos. Este lector únicamente leerá los códigos QR que haya en el museo, con lo que podemos prevenir de un funcionamiento extraño en la app o bug que entorpezca la experiencia del usuario.

Una vez que se lee el código, nos saldrá una pantalla con una imagen del personaje reconocido, su nombre, y una serie de botones:

- Botón de información: Nos dirige a la misma pantalla que nos dirigiría la app si seleccionamos el botón de ese personaje en la pantalla de home.
- Botón de más información: Este botón nos abrirá el navegador a una dirección en la que tendrá más información. En nuestro caso hemos introducido direcciones a la wiki [Hielo y Fuego](#)
- Botón de charla con el bot: Con este botón, cambiaremos la pantalla a la pantalla del bot, en la que directamente comenzará la conversación en el contexto del personaje que se ha leído en el QR.

En el último de los menús, tenemos una vista de un chat con el que podremos interactuar con el bot diseñado con la herramienta [DialogFlow](#).

Para facilitar la implementación, se ha usado una librería llamada [ChatMessageView](#), la cual nos provee de una interfaz bastante moderna y compatible con bots generados en [DialogFlow](#). El uso de esta librería también nos ha permitido que si al museo fuese alguien con problemas auditivos y/o problemas a la hora de hablar, pueda interactuar con el bot usando teclado.

También se ha creado un reconocedor de voz y un sintetizador de voz, los cuales se activan cuando pulsamos unos de los botones que están en la parte superior de la aplicación, que solo aparecerán cuando estemos en la pantalla del bot. Hay que destacar que están implementados única y exclusivamente para que funcionen en Español, no se ha probado el

funcionamiento de estas funcionalidades en otros idiomas. Para la implementación de los mismos nos hemos basado en códigos entregados y explicados en clase, los cuales hemos modificado para incrustarlos en la aplicación sin necesidad de que el usuario tenga que descargar/installar aplicaciones externas que se encargan de sintetizar y de realizar el reconocimiento de voz.

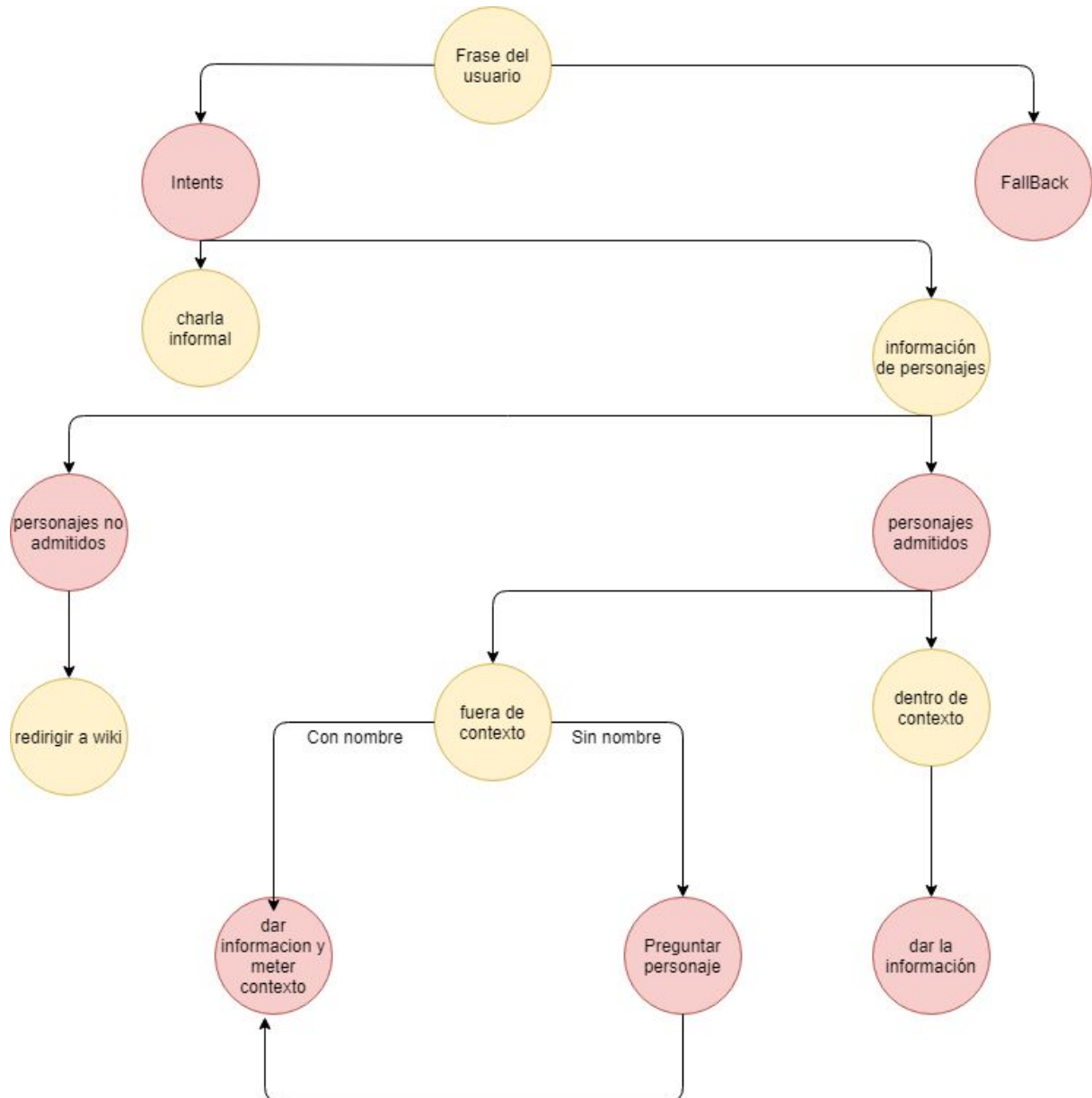
Para la parte de sensores, ya hemos explicado el primero que sería el lector de QR. Otra de las funcionalidades que se ha pensado es de tener un acceso rápido al lector de QR, ya que es una parte que hemos considerado muy importante dentro de nuestro museo. Para ello, la opción ha sido usar el sensor de proximidad. Si el sensor de proximidad detecta un cambio, en cualquier pantalla (excepto la del mapa, ya que con el zoom del mapa el usuario podría accidentalmente activarlo) y le llevará a la pantalla del lector de QR, donde podrá escanear uno nuevo. Se consideró que el funcionamiento fuese que cuando detectase el cambio activase la cámara, pero si accidentalmente se activase el sensor de proximidad, sería más lento para el usuario salir de la cámara que salir de la ventana del QR.

La última funcionalidad es la del uso del acelerómetro para mostrar un mapa del mundo de Juego de Tronos para que así el usuario pueda tener una idea del inmenso mundo de Juego de tronos. En esta funcionalidad también se ha usado el multitouch, ya que queríamos que el usuario quizás no pudiese ver bien las letras, así que se implementó la funcionalidad de hacer zoom con los gestos más habituales.

Por último todas las imágenes e información se ha extraído de la página : wiki [Hielo y Fuego](#) cuya licencia la podemos encontrar [aquí](#) y un resumen de la misma [aquí](#).

Memoria técnica de DialogFlow.

El DialogFlow está construido siguiendo este esquema:



Cuando el usuario dice una frase, puede ser que no esté contemplada por el bot, con lo cual se aclara que la frase no es viable o se va al siguiente nivel.

El bot tiene en cuenta algunos aspecto de charla informal pero no demasiado fuertes puesto que no era el objetivo de la práctica.

Si la frase va del tema de nuestro museo, de personajes de juego de tronos, ya se tiene en cuenta lo siguiente, si es uno de los personajes admitidos, se tienen varias posibilidades, de tal forma forma que para cada personaje se tiene:

- un intent “inicial” disparado en caso de preguntar quien es el personajes en cuestión, este intent dispara un contexto sobre ese personaje, y, si se preguntan las cosas contempladas que se dirán a continuación, no hará falta especificar de nuevo el nombre (followups)
- La información que se tiene de cada personaje es: los lugares en los que ha estado, las relaciones que ha tenido, las guerras en las que ha participado, y sus familiares o casa a la que pertenece.
- Se tiene también un followup del contexto del personaje cuando se le pregunta que me puedes contar o similares, en tal caso te explicará lo que te puede comentar del personaje.
- Por último un fallback que se activará si estas en el contexto y dices una frase sin lógica en el cual te explicará que o no tiene esa información o los temas de los que puede hablarte para ese personaje.
- Fuera del intent “inicial” tenemos un mecanismo que regula si el usuario quiere saber cosas de un personaje estando en el contexto de otro personaje o no teniendo contexto, por ejemplo: si se le dice al bot hablame de daenerys y justo después en que guerras ha participado Jon Nieve, en este caso el dialogflow reconoce Jon Nieve y rápidamente te da la información pedida además de poner el contexto de Jon Nieve encima del de Daenerys.

Además de esto anterior que posee cada personaje, se tiene en cuenta una serie de intents si el usuario quiere información de algo y no se tiene contexto ni se especifica nombre, en tal caso, nuestro bot preguntará a qué personaje se refiere, si la respuesta es uno de los 4 personajes, se cambiará a su contexto y se dará la información pedida anteriormente, si es uno de los personajes “no admitidos” se pasará un enlace con la wikipedia del personaje nombrado, sino no dirá nada útil y se volverá al estado inicial.

Además de todo esto si se hace una pregunta referente a uno de los personajes “no admitidos” también se redirigirá a wikipedia con el nombre de ese personaje.

Entities que se han tenido en cuenta en el DialogFlow:

- Una para cada personaje admitido que contiene sus diferentes nombres.
- Varios entities para sinonimos de palabras para facilitar el trabajo de las frases de training, estas palabras son batallas, familiares, lugares.
- Un entity que guarda todos nuestros personajes “no admitidos”

Referencias:

[Cómo crear un lector de códigos QR en Android con Android Studio](#)

[Home Icons - Free Download, PNG and SVG](#)

[Package Index | Android Developers](#)

[Exploring the Android Design Support Library: Bottom Navigation View](#)

[Android Fragment handle back button press - Stack Overflow](#)

[bassaer/ChatMessageView: Chat UI library for Android](#)

[Shake Acelerometro Sensor](#)

[Proximidad y Giroscopio](#)

[Multitouch](#)

Transparencias de clase

Apuntes de clase

Referencia de Android Developers