

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE TECNOLOGÍAS Y
SERVICIOS DE TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**DEVELOPMENT OF A MOBILE AUGMENTED
REALITY VIRTUAL ASSISTANT FOR A SMART
OFFICE**

**PABLO LOSTAO FERNÁNDEZ
JUNIO 2019**

TRABAJO DE FIN DE GRADO

Título: DESARROLLO DE UN ASISTENTE VIRTUAL DE UNA OFICINA INTELIGENTE EN REALIDAD AUMENTADA PARA DISPOSITIVOS MOVILES

Título (inglés): DEVELOPMENT OF A MOBILE AUGMENTED REALITY VIRTUAL ASSISTANT FOR A SMART OFFICE

Autor: PABLO LOSTAO FERNÁNDEZ

Tutor: CARLOS ÁNGEL IGLESIAS FERNÁNDEZ

Departamento: Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente: —

Vocal: —

Secretario: —

Suplente: —

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes



TRABAJO FIN DE GRADO

**DEVELOPMENT OF A MOBILE AUGMENTED
REALITY VIRTUAL ASSISTANT FOR A
SMART OFFICE**

PABLO LOSTAO FERNÁNDEZ

Junio 2019

Resumen

La interacción entre la máquina y el humano es un campo de estudio que ha ido aumentando en importancia gradualmente, y actualmente se encuentra en auge debido al progreso de tecnologías íntimamente relacionadas con dicho campo. Entre estas tecnologías cabe destacar dos muy ligadas a este proyecto, el procesamiento de lenguaje natural y la realidad aumentada.

El proyecto plantea el desarrollo de un asistente virtual para dispositivos móviles en realidad aumentada, con el objetivo de reproducir de la manera más natural posible la comunicación humana, un asistente al que se le pueda ver, hablar y escuchar. La función del asistente es resolver dudas sobre el funcionamiento de la oficina inteligente del Grupo de Sistemas Inteligentes (GSI).

Este proyecto se materializa con la implementación de un sistema basado en la arquitectura cliente-servidor, donde los clientes son las aplicaciones ejecutándose en distintos dispositivos móviles y el servidor es una API REST que procesa los datos relativos al lenguaje natural.

El servidor, además de dar una respuesta coherente, es capaz de extraer y clasificar los sentimientos asociados a una entrada de usuario. El cliente utiliza esa información para dotar de expresión emocional al asistente, reforzando la integridad de la comunicación.

Los resultados obtenidos cumplen con los objetivos marcados y además el servicio REST sirve de precedente para futuros desarrollos que involucren estas dos tecnologías simultáneamente, el procesamiento de lenguaje natural y la realidad aumentada.

Palabras clave: Asistente virtual, Procesamiento de lenguaje natural, Realidad aumentada, Vuforia, Dialogflow, API REST, Análisis de sentimientos.

Abstract

The interaction between the machine and the human is a field of study that has gradually increased in importance and is currently booming due to the progress of technologies intimately related to this field. Among these technologies, it is worth highlighting two closely linked to this project, natural language processing and augmented reality.

The project proposes the development of a virtual assistant for mobile devices in augmented reality, with the aim of reproducing in the most natural way possible the human communication, an assistant that can be seen, spoken and heard. The function of the assistant is to answer queries about the functioning of the intelligent office of the Grupo de Sistemas Inteligentes (GSI).

This project is materialized with the implementation of a system based on the client-server architecture, where the clients are the applications running on different mobile devices and the server is a REST API that processes the data related to the natural language.

The server, besides giving a coherent response, is able to extract and classify the sentiments associated with a user input. The client uses this information to provide emotional expression to the assistant, reinforcing the integrity of the communication.

The results obtained meet the objectives set for the project, and the REST service also serves as a precedent for future developments involving these two technologies simultaneously, the natural language processing and augmented reality.

Keywords: Virtual assistant, Natural language processing, Augmented reality, Vuforia, Dialogflow, REST API, Sentiment analysis

Agradecimientos

Agradecer a mi tutor Carlos Ángel Iglesias por su buena disposición para ayudar y a todos los profesores de la ETSIT que como Carlos colaboran a la motivación del alumno. Agradecer también a mis hermanos y en especial a mis padres, que siempre se han esforzado al máximo para darme lo mejor. Por último dar gracias a mis amigos y especialmente a mi novia que son piezas fundamentales en mi desarrollo.

Contents

Resumen	I
Abstract	III
Agradecimientos	V
Contents	VII
List of Figures	XI
1 Introduction	1
1.1 Context	1
1.2 Project goals	3
1.3 Structure of this document	3
2 Enabling Technologies	5
2.1 Natural Language Processing	5
2.1.1 Introduction	5
2.1.2 History	6
2.1.3 Concepts	6
2.2 Augmented Reality	9
2.2.1 Introduction	9
2.2.2 History	9
2.2.3 Components	10

2.2.3.1	Hardware	10
2.2.3.2	Software	11
2.2.4	Concepts	11
2.2.5	Architecture scheme	13
3	Architecture	15
3.1	Introduction	15
3.2	Overview	15
3.3	Android app	18
3.3.1	AR SDK	18
3.3.1.1	ARKit	19
3.3.1.2	ARCore	19
3.3.1.3	Vuforia	20
3.3.1.4	Comparison	20
3.3.1.5	Conclusion	21
3.3.2	Visual content design	23
3.3.2.1	3D engine	23
3.3.2.2	Character and animations creation	23
3.3.3	Text to speech and Speech to text	24
3.4	REST API	25
3.4.1	Introduction	25
3.4.2	NLP engine	25
3.4.2.1	Dialogflow	25
3.4.2.2	IBM Watson	26
3.4.2.3	Conclusion	26
3.4.3	Architecture	27
3.4.3.1	Introduction	27

3.4.3.2	Chosen tools	27
3.4.3.3	Dialogflow authentication process	28
3.4.3.4	Data processing	28
3.4.3.5	Result	28
3.5	Sentiment analysis and generation	30
3.5.1	Introduction	30
3.5.2	Specification	31
4	Case study	33
4.1	Introduction	33
4.2	User interface	34
4.3	Character representation	35
4.4	Question-answer interaction	36
4.4.1	Perceived delay	37
4.5	Sentiments expression	38
5	Conclusions and future work	41
5.1	Problems faced	41
5.1.1	The use of Dialogflow	41
5.1.2	API delay	42
5.1.3	Avatars in Unity	42
5.2	Conclusions	43
5.3	Achieved goals	43
5.4	Future work	44
	Appendix A Impact of this project	i
A.1	Social impact	i
A.2	Ethical Impact	ii

A.3 Economic impact	ii
A.4 Environmental Impact	ii
Appendix B Economic budget	iii
B.1 Physical resources	iii
B.2 Human resources	iv
B.3 Licenses	iv
B.4 Taxes	iv
Bibliography	v

List of Figures

1.1	Commercial AR assistant example	2
2.1	Intent creation screen	7
2.2	Entity creation screen	8
2.3	Intents and entities matched illustration [1]	9
2.4	AR example	10
2.5	Image target example	12
2.6	AR architecture scheme in smartphones. Based on [2].	13
3.1	Client-server architecture	16
3.2	System architecture	18
3.3	Relative number of devices running a given version of the Android platform [3]	22
3.4	Example of how Dialogflow handles a user utterance [1]	26
3.5	API architecture	27
3.6	BML body behaviour [4]	30
4.1	App logo	33
4.2	User interface	34
4.3	Character representation	35
4.4	Welcome intent interaction	36
4.5	Delay in the conversation	37
4.6	Sentiment expression	38

Introduction

1.1 Context

The interactions between humans and machines are technologically increasingly complex but, in exchange, they are increasingly natural for humans. Nowadays, the evolution of these interactions is an important field of study that includes many technologies that have experienced great advances in the last years [5].

Big Data, Artificial Intelligence, Machine Learning, Natural Language Processing, Augmented Reality, Virtual Reality and Mix Reality are some concepts that have been acquiring fame in the society, all of them, in one way or another, related with this project and to each other. Augmented Reality and Natural Language Processing are addressed in depth in this development because they are the closest ones to the final result, the augmented reality virtual assistant.

Natural Language Processing (NLP) is a sub-field of artificial intelligence that, as the name suggests, studies how the computers process and analyze human language data. It is behind the assistants that have been popularized, making people's life easier. Some examples are Siri, Cortana, Alexa, Google Home and many others with different applications, for instance in customer service [6].

Digital realities have been popularized too, especially the Virtual Reality and the Augmented Reality. Virtual reality (VR) is an artificial and computer-generated recreation of a real-life environment or situation. It immerses the user by making them feel like they are experiencing the simulated reality firsthand, primarily by stimulating their vision and hearing. Virtual Reality is leading the future of video gaming, and the future of the newest learning techniques, mainly in the most expensive training like military, medicine or pilot learning. The really important digital reality for this project is Augmented Reality (AR) that adds digital layers to an existing reality in order to make it more meaningful through the ability to interact with it. Augmented Reality is having a considerable impact on learning methods, preview tools for architecture, interior design, clothes shops, and improving the user experience in the interactions with the computers [7].

In this context, combining these two technologies of growing importance (AR and NLP), the augmented virtual assistants appeared. They are virtual assistants that include visual interaction and can be virtually placed in the real world using a camera and a display. They can use written or oral communication and they are able to converse about a definite matter. These assistants are been used in shopping centres, big events or companies.

There are many options to develop augmented reality and natural language processing applications and it is necessary to select the most appropriate ones for the objective, so first of all, this objective must be specified.



Figure 1.1: Commercial AR assistant example

1.2 Project goals

The objective of this project is the integral development of an augmented reality virtual assistant for mobile devices. The assistant has to be able to hold a conversation about the GSI smart office, about how it works. Below, all specific aims are listed, and they are some desired requirements of the final result.

- Develop an Android app compatible with most devices.
- The app has to use augmented reality to represent an animated character.
- The app has to simulate user-character communication.
- The character has to be able to express emotions.
- Communication has to be preferably oral.
- The project must help future NLP developments with Unity.

1.3 Structure of this document

In this section, a brief overview of the chapters included in this document is provided. The structure is as follows:

Chapter 1 Introduction This chapter helps the reader to understand the context and the project purpose.

Chapter 2 Enabling Technologies It is a deeper analysis of involved technologies.

Chapter 3 Architecture This chapter describes all technical features of the project including problems and solutions selected.

Chapter 4 Case study In this chapter most interesting use cases are shown with real images of the final result.

Chapter 5 Conclusions and future work This chapter summarizes the conclusions and checks if the result obtained is coherent with the result expected. It also looks at the future of the project, repairs and improvements.

Enabling Technologies

2.1 Natural Language Processing

2.1.1 Introduction

First of all, it has to be explain that terminology is often changed on the Internet. Sometimes people talk about NLP only like a part of communication between human and machine and it uses natural language understanding (NLU) and natural language generation (NLG) to complete the communication. Other times NLP includes everything, and NLU and NLG are parts of it. In this project, when NLP is mentioned it is referring to the second interpretation, the processing task includes the understanding and generation ones.

Like it is mentioned in the project introduction, Natural Language Processing is a sub-field of artificial intelligence that studies how the computers process and analyze human language data, however, this definition is not complete at all, mainly because it has evolved a lot and nowadays, it includes the natural language generation. Machines are not only capable of taking human language data and process it to make mathematical operations to draw conclusions, but they also are capable to answer coherently, understand conversation contexts, even computers are able to interpret data to understand the interlocutor senti-

ments. That is a great step to improve human-machine interaction, the main reason for this project [6].

2.1.2 History

NLP is not different from the most famous technologies, it is not new, however, it has suffered great improvements during this century. It is considered that NLP appeared during the fifties when Georgetown University and IBM developed an experiment that consisted of the automatic translation from Russian to English. It was during the eighties when there was a revolution in NLP with the growth of machine learning and statistical models that make probabilistic decisions based on real data. It has evolved jointly with machine learning, with unsupervised and semi-supervised learning algorithms (algorithms that are able to learn from data that has not been hand-annotated by a human, for instance from the world wide web content). During this decade, at least for now, the final leap has taken place using the newest machine learning techniques, in particular, the feature learning (an algorithm that permits a system to automatically spot how to detect and to classify certain features from raw data) and deep learning (based on artificial neural networks).

2.1.3 Concepts

NLP is the technology behind the assistants but to develop these assistants the commercial products that do not require machine learning knowledge are used. They are a reasonable solution for developers to access NLP engines. One of these tools to offer NLP solutions is chatbot-based software. Dialogflow, IBM Watson and many other examples are solutions to use the most powerful advances in NLP using chatbots, they allow clients to build their own bots that are able to talk about a specific topic. All of them have their distinctive features, some offer more customization, others offer simplicity. However, there are some common terms in these solutions and it is important to understand them because these terms are constantly used when a bot is developed.

These solutions understand a conversation as a group of intentions (the user purposes), entities (classifiable information) and relations between them (contexts). These three elements are enough to create conversations and they have to be manually created by the bot creator. These concepts are explained in more detail below [1].

- **Intent:** It is a user intention. It does not represent one way to say something, it represents all the ways of saying that something. For instance, the sentences “Do you

know what time it is?”, *“What time is it?”*, *“Have you got the time?”* and *“Could you tell me the time?”* have the same purpose, the speaker wants to know the time. Each intent represents a user purpose, and they have to be defined by the bot creator, specifying at least the name, training phrases and possible answers.

In the next image, it is shown how intent is created. *Default Welcome Intent* is the name of the intent and it has some training phrases and some responses. This training phrases that have been put manually are used by the NLP engine to deduce new ways to express the same intent. When this intent is matched (it is considered that the user has this intention), the bot will answer one of the responses randomly.

• Default Welcome Intent SAVE ⋮

Training phrases ? Search training phrases Q ^

” Add user expression G

” buenos días

” saludos

” hola

Action and parameters ∨

Responses ? ^

DEFAULT +

Text response ? 🗑

- Hola, bienvenido. ¿En qué puedo ayudarte?
- Buenas, me alegra verte. ¿Qué puedo hacer por tí?
- Buenos días, ¿Qué necesitas?
- Enter a text response variant ⬇

ADD RESPONSES


Figure 2.1: Intent creation screen


- **Entity:** It is a classification of relevant information in the sentences. Entities permit to extract particular information that belongs to a type of word. For instance, *“fruit”* can be an entity and when the question *“What is your favourite fruit?”* is asked, the answer can be *“My favourite fruit is the strawberry”*, so in this case, the word *“strawberry”* is associated with the entity *“fruit”*. It is said, that *“fruit”* is an entity

and “*strawberry*” is an entity entry, that can have more than one word for the same thing. “*Sweet potato*” and “*yam*” are two different ways to refer to an entity entry that belongs to an entity (tubers).

This image illustrates the creation of the entity *Programming Language*. It has many entity entries, one for each language and for example, JavaScript has JS as a synonym.

programmingLanguage SAVE

☒ Define synonyms  ☐ Allow automated expansion

 Separate synonyms by pressing the enter, tab or ; key. ×

JavaScript	JavaScript, JS
Java	Java
Python	Python
C#	C#
C++	<input type="text" value="C++"/> Enter synonym

Figure 2.2: Entity creation screen

- **Context:** Contexts represent the particular states of a conversation and allow the bot to carry information between intents. For example, if someone asks “*Why?*”, this question does not make sense without context. Like this, many interventions need a context to be answered. Following the example above, if “*Why?*” is an answer to “*My favourite fruit is the strawberry*”, the final answer has to be coherent with the flow of the conversation, the answer has to explain why strawberry is the favourite fruit and not why else.

In the following image, there are three examples that show three ways to ask for forecast, in one of them there is no an entity associated, in other, there is one entity (time) and in the other, there are two entities (time and location). Entities are used to carry information from the user input to the answer but they also are used to do actions in external apps. For instance, in the image below the entities *Time* and *Location* can be used to make a request to a external weather API (Application Programming Interface).

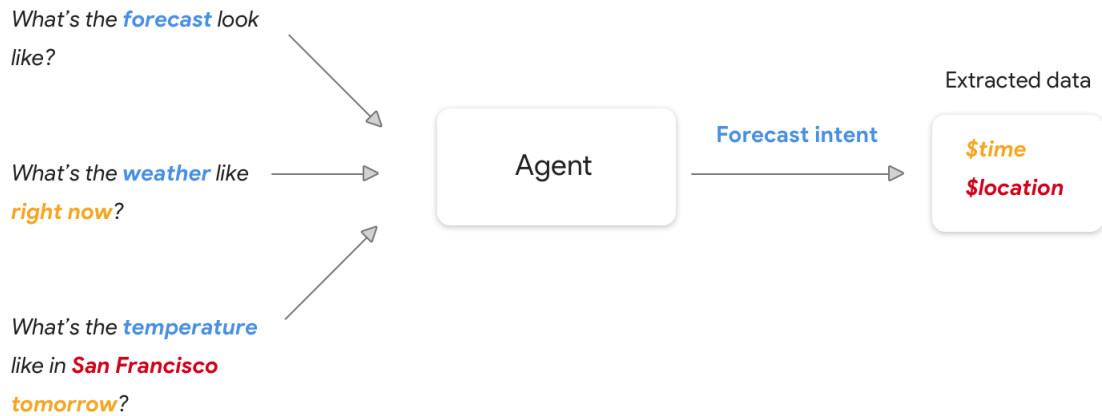


Figure 2.3: Intents and entities matched illustration [1]

2.2 Augmented Reality

2.2.1 Introduction

Augmented reality is one of the three digital realities that are changing how humans sense virtual content, making this experience more realistic. AR consists of adding digital information to the real world interpreting it through a camera and sensors, processing virtual content and representing it through a display. One clear example of this is the increasing tendency in TV that uses AR to show graphics for spectators like can be seen in the image in the next page.

In this example, there is a camera recording the scene and there is a computer interpreting it. The computer is capable to know where the horizontal planes are, where it has to put the processed information and other relevant data about the environment. Once the computer has understood how the scene is formed, it adds to the original image the corresponding graphic [7].

2.2.2 History

AR appeared during the sixties and it is believed that the first AR device was invented by Ivan Sutherland in 1968, but it is not clear if that device was closer to virtual reality. In 1974 Myron Kruger developed his project called “Videoplace”. That project combined a projection system and many cameras to produce shadows on the screen, making an interactive environment. The term “augmented reality” was coined during the nineties, and two years later the first real operational AR system was created by Louis Rosenburg.

At the beginning of the century, the AR technology changed, it stopped being a technology for the most powerful companies when Hirokazu Kato (Nara Institute of Science and Technology) released ARToolKit, a software that was going to expand AR to the rest of the world. Today AR is part of the society. It is commonly known in TV but it is used in video gaming, architecture, education and many other fields.

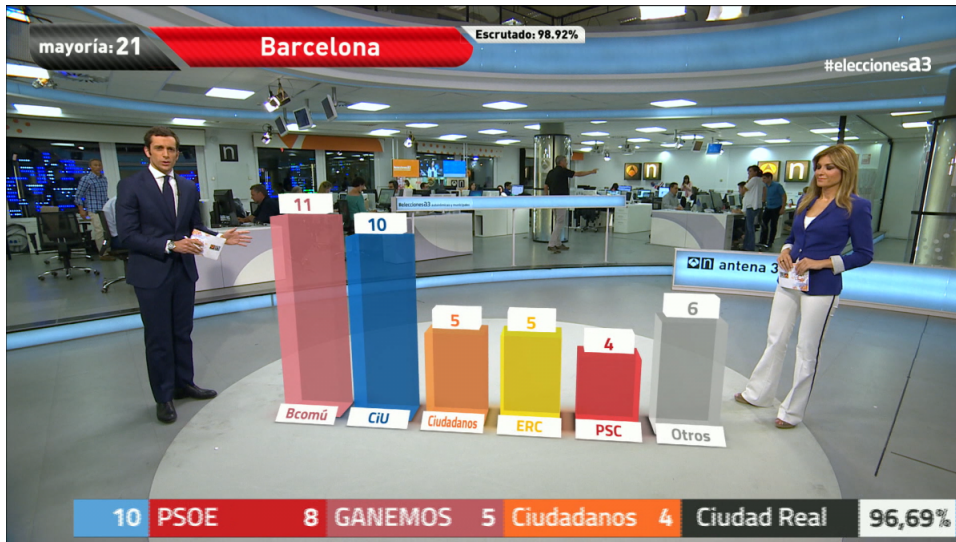


Figure 2.4: AR example

2.2.3 Components

2.2.3.1 Hardware

AR needs at least a camera, a processor and a display. Nowadays all these components are inside any smartphone, but a computer with a camera is equally suitable. One of the advantages of using smartphones is that they include many useful sensors for AR, for example, the accelerometer, the gyroscope, or the compass. However, smartphones also have weaknesses, mainly processing power. Even though mobile devices have increased their power considerably, the computing capacity is not comparable.

A display is needed, but it has not to be a conventional display, in fact, there are some alternative options for the visualization of the image, like eyeglasses, head-up displays (HUDs) or even the virtual retinal display (VRD), that is been developed at the University of Washington [8].

2.2.3.2 Software

AR uses some computer vision methods to analyse the scenes, and these methods have been inherited from the advances in robotics to determine the position of a robot, known as visual odometry. These methods usually work in two stages, firstly they detect interest points, objects with known geometry and the pattern of apparent motion of objects and surfaces (optical flow), and secondly they try to restore the real world coordinates from the data from the previous step. AR Markup Language (ARML) was developed based on Extensible Markup Language (XML) to describe how the virtual objects are integrated into the scenes [9].

Above, it is explained superficially how the AR software core works, but to develop AR application it is necessary to work with tools that implement all these functionalities. With ARToolKit as a pioneer, today there are many options to develop AR for all devices and operating systems. These tools are known as Augmented Reality Software Development Kits (AR SDKs) and they are the starting point to create final user applications. Technically SDK and Toolkit are not the same, an SDK is a complete group of development tools that is used to develop for a specific environment, and a Toolkit only enables some functions for apps running on concrete platforms, however, it is common to compare between them and not to stress the differences because in certain levels they overlap. This has to be clear because in this text the word SDK is used to refer properly SDKs but Toolkits too. In the next section, the main concepts that these SDKs use will be explained.

2.2.4 Concepts

It is needed to clear some AR concepts that are important to understand the differences between SDKs and how the app is working [10].

- ***Fiducial marker:*** It is an object placed in the field of view that is used as a point of reference. The most basic AR experiences need one type of these markers to place virtual content known as *target*. The target usually is easily recognized by the AR engine, requiring few computing resources. Corners, contrast, characteristic lines and geometrical figures are some resources to make readily recognized targets. Like it has been emphasised, this is only for the most basic AR experiences, but targets can be more complex, like spheres, cylinders, or directly planes, however, the needed computing capacity grows exponentially.

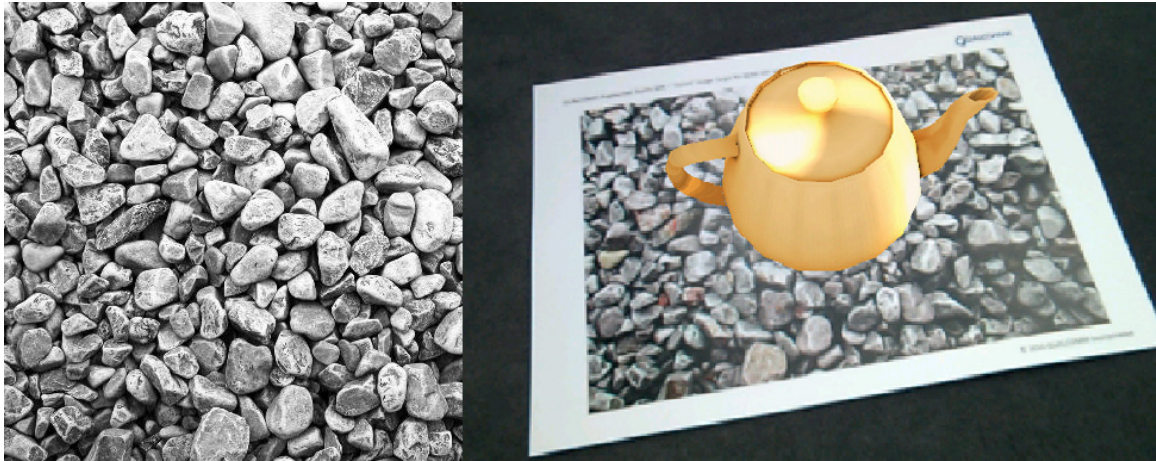


Figure 2.5: Image target example

The images above show how the image target works. The image on the left is the image target and the image on the right is an AR scene. The target is printed and it is put in the real world. When the scene is captured by the camera, the processor recognizes the target and it knows where it has to put the virtual content, in this case, the cup.

As can be seen, this target has many characteristic points and a high contrast. This facilitate the recognition to the processor. It is one of the most important things regarding the performance of augmented reality.

- **Tracking:** It is the process by which the content is attached to a fixed point in space. When targets are used, the recognized points are these fixed points where the content is attached, and when there is no target (for instance, the content is put in horizontal planes) is needed to use other mapping technique (usually SLAM).
- **SLAM:** SLAM means *Simultaneous Localization and Mapping* and it is a technique to build a digital map of an area. It has been a great advance in AR because it permits to affix virtual content in whatever point of the space and to recover it in subsequent sessions.
- **Six Degrees of Freedom (6DoF):** This term refers to the range of motion in relation to virtual content. Three of these six degrees allow the user to move through the three perpendicular axes in the space and the other three permit the user to rotate through those axes.

2.2.5 Architecture scheme

In this image, it is shown a scheme about how components interact between them to create the AR scene in a smartphone, following the described foundations. Sometimes, the process is whole in local, as in the image, but other times the rendering engine communicates with a web server that has the virtual objects manager and the virtual objects database [2].

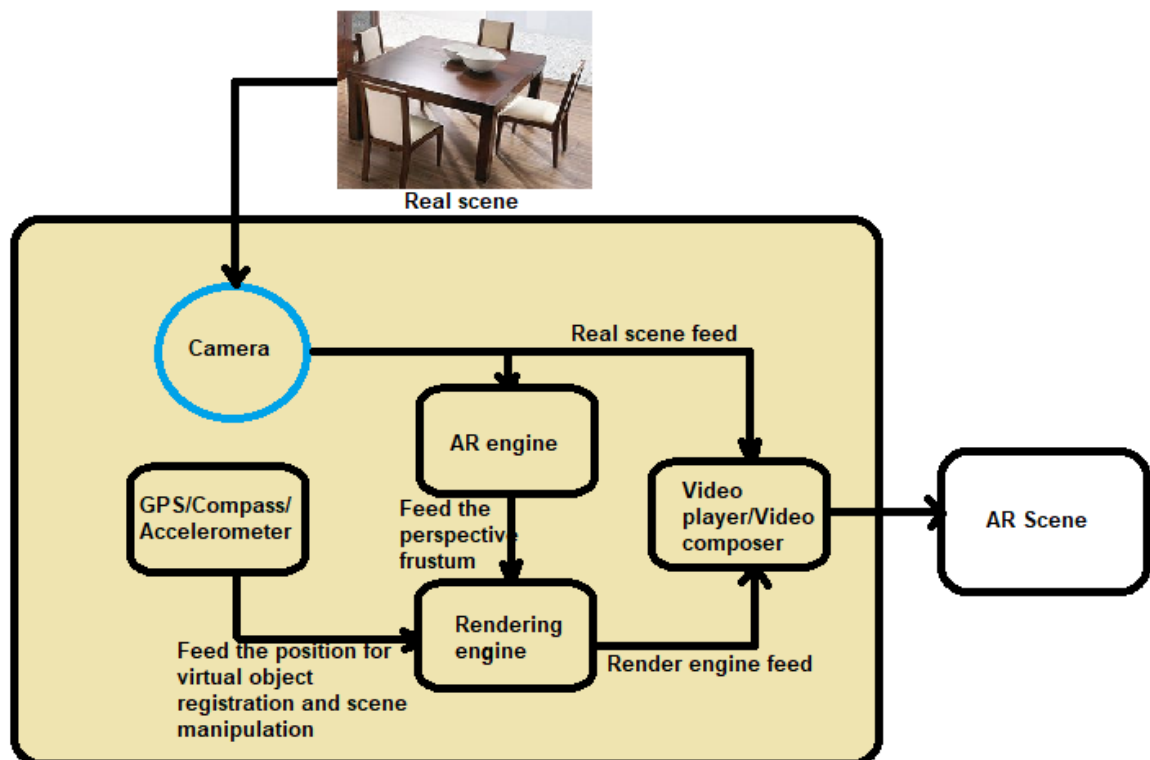


Figure 2.6: AR architecture scheme in smartphones. Based on [2].

Architecture

3.1 Introduction

In this chapter, the design phase of the project is covered, as well as implementation details. First of all, an overview of the project is presented and secondly, the different components are analysed in depth. The intention is to choose the most appropriated pattern to the objective and to describe how that pattern is implemented.

3.2 Overview

There are many different architectural patterns in software design. All of them are general solutions for common problems in software architecture. Layered, client-server, model-view-controller are some examples of these patterns. For this project, the most appropriated one is the Client-server pattern [11, 12].

The client-server pattern is very known in software development, and it consists of a server that offers services to many clients. The server is listening, and the clients can request these services to it. This pattern offers some advantages that are very useful for this project.

- **Centralised control:** Data is managed by dedicated servers, so it is easier to make changes in the system and to control the access to these data.
- **Scalability:** This pattern permits to increase the number of nodes independently.
- **Encapsulation:** It allows to abstract the logical functions, hiding information to higher level objects.

Applying Client-server model to the project, the Android applications are the clients that are going to delegate NLP functions to the server like it is shown in the next image. The system measurement is necessary to ensure the correct functioning because all requests made in the different mobile devices are managed by a determined number of servers. In this project the number of clients operating at the same time is reduced, so one server is enough to give service to all clients.

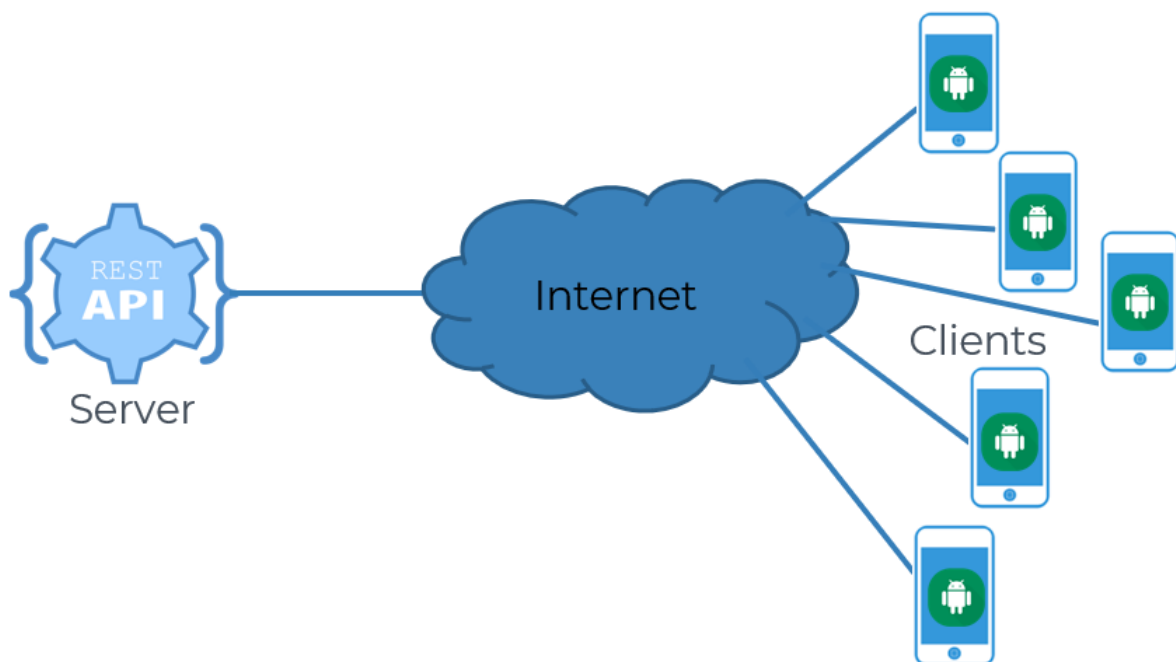


Figure 3.1: Client-server architecture

The communication method between client and servers is object of study too, and it can be seen that in the image the server is specified as REST API. First of all, it is needed to explain what is REST and its characteristics. REST (Representational State Transfer) is the most used software architecture technique to communicate servers and client. Some features have to be explained to understand the utility of this technique [13].

- ***It is simple.*** REST is an alternative to build APIs to SOAP(Simple Object Access Protocol), and the main advantage is that its implementation is much simpler.
- ***It uses HTTP protocol.*** REST is based on HTTP (Hypertext Transfer Protocol), and it uses the HTTP verbs for requests (GET, POST, PUT and DELETE). Studying REST services or HTTP, there are many people that associate REST (or HTTP) verbs with CRUD operations (Create, Read, Update and Delete) but this comparison is not very suitable, especially POST like it is deduced of this work. In chapter 5, some problems faced are described and one of these problems helps to understand why there is no rigorous equivalence between the HTTP verbs and the CRUD pattern.
- ***Stateless.*** Each transaction is independent and it does not have any relation with past requests. The server does not save information about sessions.
- ***Universal syntax.*** Resources are identified by unique URIs (Uniform Resource Identifier).
- ***Formats.*** Data is handled with XML and mostly JSON (JavaScript Object Notation) because it is lighter and consequently more useful.

Once the REST concept has been understood, an API REST is an application programming interface that uses that architecture.

Knowing all this, the general functioning is easy to specify. Android apps (clients) have to delegate NLP functions making a request with relevant data to an API REST that is going to process them. At the same manner, API REST sends the response with processed data to each client. The app has to handle the character representation, to send the user input data to the API and to show the answered data to the user.

All this process is represented in the following image, where the interactions between different modules are shown too.

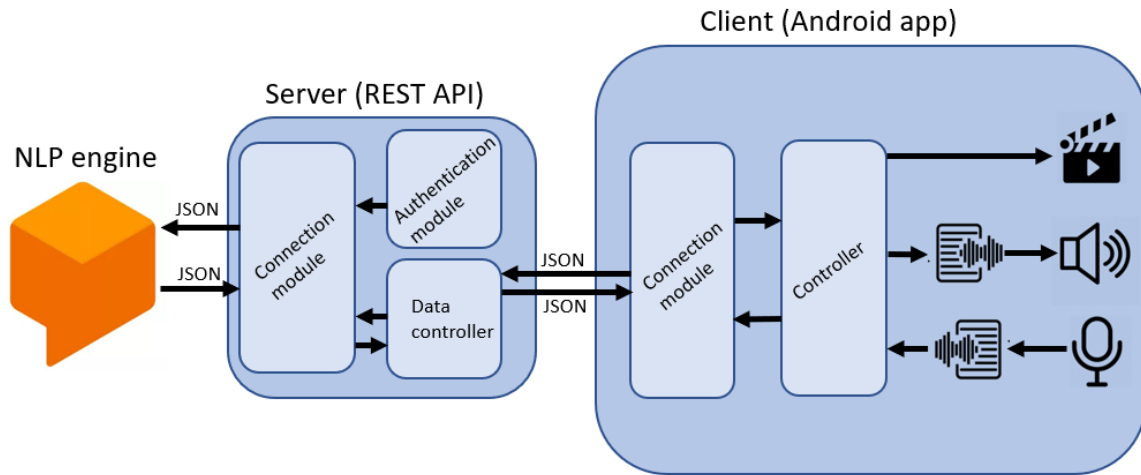


Figure 3.2: System architecture

The aim of this overview is to help the reader to clarify the working of the system before the deep analysis of the components that is exposed below.

3.3 Android app

Android development is one of the keys of this project but in this case, it is very conditioned by augmented reality. Like it was mentioned, there are many AR SDKs that are different and are going to condition the complete development. Firstly it is necessary to choose the most appropriated one for the specific application, making a comparison between the most important ones.

3.3.1 AR SDK

There are many options and all of them have some pros and cons, and there is no one that is the best in every aspect. However, there are three SDKs that can be considered more important than the others, principally because they have the biggest communities behind them. For this reason, it has been decided to analyse in depth only three choices. These three options are ARKit, ARCore and Vuforia. The importance of ARKit and ARCore lies in their developers, they have converted their SDKs in one of the most powerful tools for AR development in a short time. ARKit is developed by Apple, and ARCore is developed by Google. Vuforia was created by Qualcomm Incorporated and it was bought by PTC in 2015. Now Vuforia is integrated into Unity (the 2D and 3D design engine) so this is a big advantage to create AR experiences for hand-held devices [14].

3.3.1.1 ARKit

ARKit is a relatively new SDK, it was launched in June 2017. It only works with Apple devices but now it is one of the best tools to work on augmented and virtual reality because it stands out among the others because it is very simple. Really, ARKit is not valid for this project because it does not work with Android, but it is interesting to include it in this analysis because it is probably the most important SDK nowadays. The first version was launched for iOS 11 onwards, and the second one was launched only for iOS 12 onwards. The first version included 2-dimensional image recognition and tracking, but it also included the capacity to interpret 3-dimensional spaces, surfaces and objects making it possible to place virtual 3D objects in the scene. One year later, in June 2018 Apple launched the second version (ARKit 2) that started detecting and tracking 3D objects. This new version introduced two new characteristics that have been the key to improve AR experiences:

- **Persistence:** The persistence is the feature that allows you to resume an old session, for example, if you were playing an augmented reality board game, you can stop it and continue the game as you left it.
- **Shared experiences:** The AR experiences are not limited to a single person, you can play with other people interacting in the same space.

This SDK is free to use but there is a developer license for distribution that costs ninety dollars a year.

3.3.1.2 ARCore

ARCore was published at the beginning of 2018 in response to Apple. In a first moment, Google planned to develop the SDK only for Android devices, as Apple had done, but finally, ARCore works with Android devices (Android 7.0 and higher) and with Apple ones (iOS 11 and higher). Google does not have advances as interesting as Apple ones in shared experiences, but they have put their effort to merge the virtual and real world with the maximum possible perfection. This is possible on account of these three capabilities:

- ***Motion tracking:*** It is the capability that permits the devices to understand their relative position in the world through precise sensors.
- ***Environmental understanding:*** It is orientated to understand the location, size and shape of the real objects and surfaces in the scene, and to integrate virtual elements with more sense and precision.
- ***Light estimation:*** It makes possible to understand the real-life lighting conditions to improve the quality of the understanding and the painted virtual objects

ARCore is completely free to use.

3.3.1.3 Vuforia

Vuforia is one of the most famous tools to work with augmented reality. It was born in 2010 and it supports Android, iOS and Windows and it has had the biggest community of developers because it was one of the firsts, because it is free to use and because always it has been at the top in functionalities. Vuforia implements an object scanner that can create virtual objects from real things. For this creation, it uses a database (local or cloud). Another important aspect of Vuforia is that it is totally integrated with Unity, the most important engine to create 3D and 2D graphics, and through Vuforia Ground Plane it is possible to create virtual scenes with a better interpretation of the planes and other surfaces. From Vuforia 7.2 it supports ARCore and ARKit, so on compatible devices, Vuforia is able to leverage the best characteristics of the others SDKs.

Vuforia is free to use, but it includes a watermark and a limit in cloud recognition.

3.3.1.4 Comparison

Now that an overview of each analysed SDK has been made, in the next table, a comparison of relevant features is shown.

	ARCore	Vuforia	ARKit
Compatibility with Android	Yes(>7.0 Nougat)	Yes(>4.4 KitKat)	No
Compatibility with iOS	Yes(>iOS11)	Yes(>iOS7)	Yes(>iOS11)
Unity 3D support	Yes	Yes	Yes
Cloud recognition	No	Yes	No
3D Tracking	Yes	Limited	Yes
Smart glasses support	Yes	Yes	Yes
Licence type	Free	Free(Watermark) Commercial	Free
Degrees of Freedom	6	6	6
SLAM	Yes	No	Yes

There are so many features so the most representative ones for our project have been considered.

3.3.1.5 Conclusion

As seen above, all of these tools have some unique features and there is no one perfect. This comparison is not absolute and it is not transferable because it has been done considering the specific requisites of this project. Firstly, it has been considered the importance of the compatibility with the people's devices, and it is clear that Vuforia is the winner at this point:

In Europe, Android has a 70.91% of market share and iOS 27.95%. In addition, the Android market is fragmented in this way:

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%

Figure 3.3: Relative number of devices running a given version of the Android platform [3]

It is clear that Vuforia has the best compatibility. ARKit only works with iOS devices that has at least iOS 11 version, and the oldest device that have been updated to iOS 11 is iPhone 6, so ARKit only can be compatible from this iPhone onwards. ARCore works with iOS devices, like ARKit, and in Android devices, only if they have at least Android 7.0 Nougat version and they are in a select list of high-quality phones. ARKit and ARCore have been designed to understand the real world only in a very demanding way, so it is required to have one of the most powerful phones to use them. The use of ARKit is dismissed because the development of the application only for iOS devices is not considered.

Comparing Vuforia with ARCore it is more difficult to choose one of them. ARCore has better integration in Android Studio than Vuforia and it has some advanced features that signify a great increase in the potential of the application (motion tracking, environmental understanding and light estimation).

However, considering everything, it has been decided to use Vuforia AR SDK because in addition, from 7.2 Vuforia version it can work with ARCore in supported devices to implements the features that were mentioned before, so it makes possible to develop an application with high compatibility and then, to develop another version with some extra features.

3.3.2 Visual content design

3.3.2.1 3D engine

There are many 3D engines, like Unity, Unreal or Godot. The first two are the most used ones and both are free for this project. Unity is free for personal use and Unreal is free if benefits are less than 3,000\$ per product. This project does not have complicated scenes, so both engines can achieve the purpose and more. For this reason, the engine is chosen considering the relation between Vuforia and Unity. Unity has facilities to develop AR apps using Vuforia so it is going to be used.

One of the best advantages of Vuforia is the complete integration with Unity 3D. Unity is a very important development engine to create three-dimensional content and it is very used in video gaming to animate characters. Now, Unity has Vuforia integrated, so it is possible to develop AR applications for many platforms using only Unity 3D. Unity can export a project for Android Studio, but it can generate directly the APK file too. Unity works with C# that is an object-oriented programming language standardised by Microsoft.

Unity has its own store where the developers purchase extensions, libraries, avatars, animations and other assets. An asset is the representation of any item that can be used in a project. There are assets created outside Unity that are compatible with the engine, for instance, audio files, images and 3D models [15].

3.3.2.2 Character and animations creation

In the Unity asset store, there are many assets like characters or animations, however, the free ones have very reduced quality. After some attempts that will be explained in the following chapters, a great solution to create avatars and animations were found. It is a group of specialised 3D animation software developed by Reallusion Inc. Three of these programs are used, Character Creator, iClone and 3DXchange.

- *Character creator:* It is a software to create characters, specialised in avatars. It

is considered an avatar a character that is formed like a human. Avatars are very important because their animations can be exchanged.

- ***iClone:*** It is a real-time 3D animation software that helps the developer to create professional animations for films, video gaming, previews or like in this case, AR and VR content.
- ***3DXchange:*** It is a tool to export all types of 3D assets, and it is used to export the assets created with iClone and Character Creator to Unity.

3.3.3 Text to speech and Speech to text

The complete development is made using Unity, so it is needed to add Speech to text (STT) and Text to speech (TTS) functions to the app using Unity. A free Unity plugin called *Android Native Dialogs and Functions Plugin* has been used.

3.4 REST API

3.4.1 Introduction

In the overview, it has been explained what a REST API is, but now, it is required to emphasise the concept of an API. It is the formal specification about how a software module interacts with others. For this reason, understanding it independently, the REST API is the most valuable and complete product of this project. It is going to provide an easy way to communicate AR Android applications made with Unity with one of the most advanced NLP engines.

The necessity of this API resides in the difficulty to find native Unity package for NLP engines. For instance, Google only has the first version of Dialogflow (called API.AI) as a Unity package and the IBM solution, Watson, is no longer available. Finally, after considering some other options explained in the following chapters, it was decided that the easiest way was this API, a bridge between the Android app and the NLP API.

3.4.2 NLP engine

In this section, some NLP engines are compared to decide the most appropriated for the project, like it was done with AR SDKs. Like in that case, there are many alternatives and it does not make sense to make a comparison with all of them. Two of them mentioned above have been chosen, Dialogflow and Watson that are the Google and IBM solutions respectively. Both are very similar, so the differences are summarised.

3.4.2.1 Dialogflow

Dialogflow is the newer version of API.AI that was not owned by Google originally. This first version was released to third-party developers in 2014 and it already included voice recognition and text to speech. Google purchased API.AI in 2016 and it was used to help developers to create new actions for Google Assistant.

Today, Dialogflow supports more than 14 languages and it works with text and voice. It implements the best Machine Learning techniques, and that makes Dialogflow the best understanding of the context of the conversations. Dialogflow includes sentiment analysis and it is very easy to get started. The next image illustrate how Dialogflow handles a user utterance, and it is very similar in Watson.

The standard version is completely free and it covers all the needs of this project.

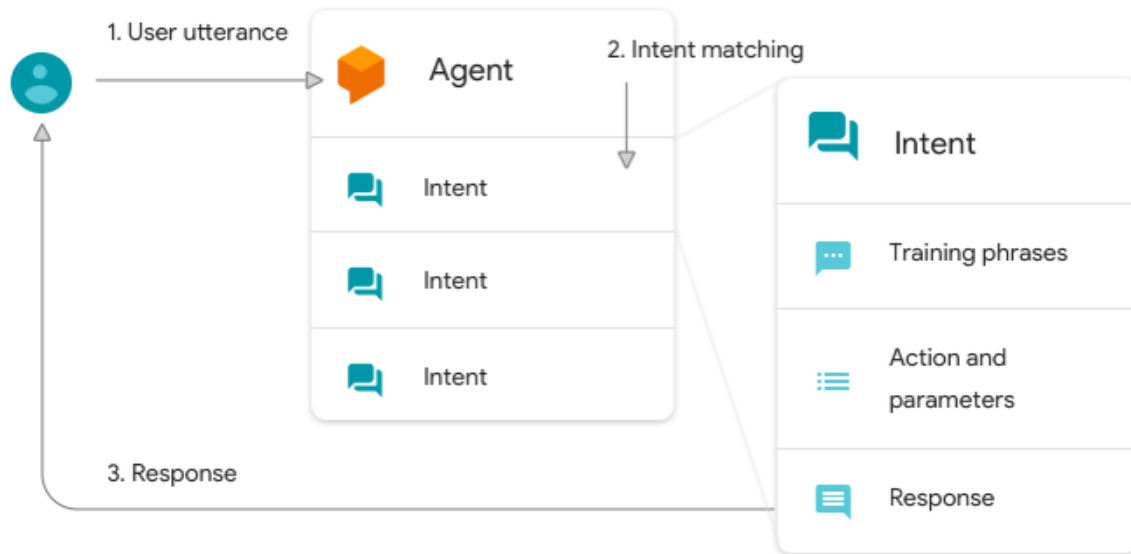


Figure 3.4: Example of how Dialogflow handles a user utterance [1]

3.4.2.2 IBM Watson

IBM Watson was initially developed by IBM to answer questions on an American television game show called Jeopardy and that goal was achieved in 2011 when the AI system won the first place prize, one million dollars. In 2013, the first commercial product was released.

One of the greatest advantages of Watson is its simplicity in complex developments, however, the initiation phase is less intuitive. It offers a very high level of conversation flow options based on a hierarchical structure. The conversation design is more natural. Watson is also able to work with sentiment analysis, even it is capable of analysing the voice tone.

IBM Watson offers a free plan that has a maximum of 10,000 messages per month and some limitations regarding engine capabilities.

3.4.2.3 Conclusion

Both IBM Watson and Dialogflow are suitable for the project but finally, Dialogflow has been chosen because it does not have any limitations in functionalities and it does not have limitation in the number of messages either.

3.4.3 Architecture

3.4.3.1 Introduction

In the previous section, it has been decided which NLP engine the API is going to work with but actually, it is external to the API. In this one, it is explained how the API works and the selected tools to develop it. It is detailed how the authentication is made too.

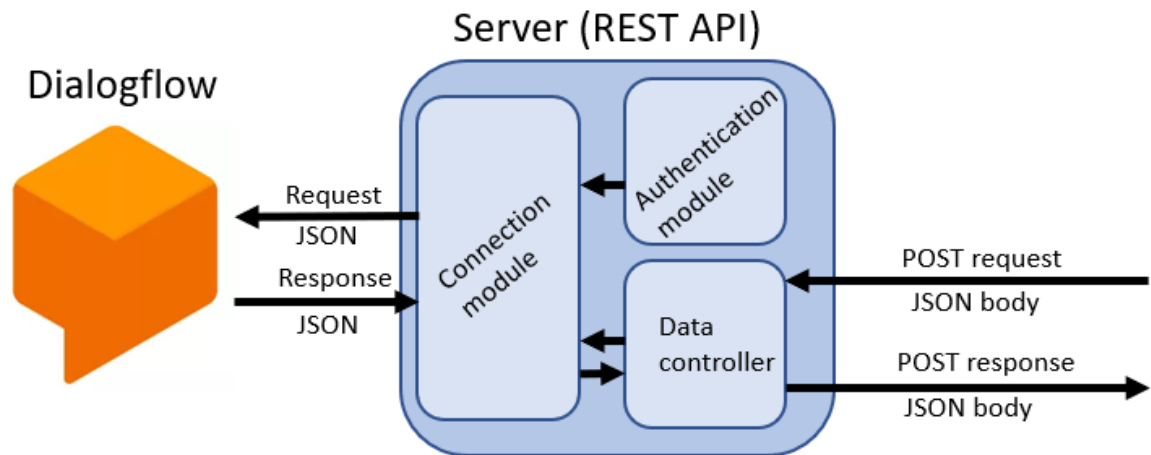


Figure 3.5: API architecture

3.4.3.2 Chosen tools

This is not a complex API and it can be developed with many languages and the result is not significantly different. For this reason, chosen tools are not a product of a deep analysis, the decision has been taken following personal preferences.

JavaScript, Node, NPM, Express and Heroku have been selected and below it is explained which function performs each tool.

- **NodeJS:** It is the JavaScript runtime environment, that is, it offers an environment where standalone JavaScript applications can be executed, without a browser interpreting it.
- **NPM:** It is a package manager for Node that makes easier to manage the modules of the application. It is an easy way to manage project dependencies.
- **Express:** It is an open-source web application framework for Node designed to build web applications and APIs.

- **Heroku:** It is a cloud Platform as a Service that permits companies to build, deliver, monitor, and scale apps. The API is deployed on Heroku.

3.4.3.3 Dialogflow authentication process

The first version of Dialogflow used an authentication method based on access token, but it has been exchanged for one based on Service Accounts. It consists of creating a Service Account on Google Cloud Platform and using the private key associated with that account for the authentication.

The private key is downloaded as a JSON file that is used to make authentication requests. The file has to be part of the API file structure [16].

3.4.3.4 Data processing

The API receives POST requests with a JSON structured body and first of all, it extracts the field `inputData` that contains the text converted from the input user audio. From this extracted text the API builds a Dialogflow request and sends it.

When Dialogflow response arrives the API it extracts relevant information, concretely, the text answered, and if any sentiment has been analysed. Then, it builds the final response for the Android app and sends it.

```
{
  "dataResponse": "The first thing you have to do is to send an email to the
    administrator so he can register you, indicating your username and ID number.
    Once registered you can access with your ID.",
  "emotion": "No sentiment Analysis Found"
}
```

This is the JSON of the response and in this case, the emotion field specifies that there are not sentiments found. This is because the input data is only *How can I use the electronic ID?*.

3.4.3.5 Result

One of the most important things building an API for real-time conversations is the timing. If the API that does not accomplish a reasonable delay it is completely useless. For this reason, it is needed to ensure that the API is fast enough.

Average response time has been calculated from a sample of 50 requests made from a REST client and with different intents excluding the first request. The first request is excluded because Heroku apps are usually asleep and it is with the first request when the app awakes. The average is 972.951ms and 1204.59ms and 749.36ms the maximum and minimum respectively.

With the intention to give rigorous meaning to the numbers above, those numbers are compared with Robert Miller's study [17] that has served as a reference since it was published in 1968. This research maintains that there are three different ranges in how humans perceive response speed.

- ***Less than 100ms:*** It is perceived as instantaneous.
- ***Around 1s:*** It is fast enough for users to feel they are interacting naturally with the information.
- ***More than 10s:*** It loses the user's attention.

The study concludes that 2s is a reasonable limit for real-time communications, so if this conclusion is translated to this project, it can be ensured, the API is able to simulate real-time conversations.

3.5 Sentiment analysis and generation

3.5.1 Introduction

This section is separated from the API section and the app section because it is not part of them, the sentiment analysis and generation is a process that needs both parties. This process has been created during the project based in Behaviour Markup Language (BML). BML is a language that describes human nonverbal and verbal behavior in a manner independent of the particular realization (animation) method used [4].

In the figure 3.6 it is shown a representation about the BML body.

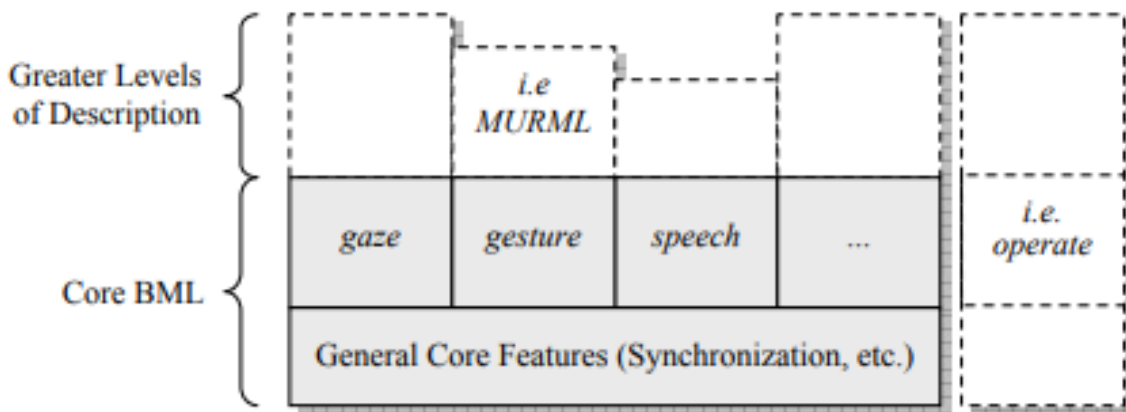


Figure 3.6: BML body behaviour [4]

It includes the BML core and the greater level of description which makes possible to describe in more detail the BML attributes. One of the main functions of the core is to synchronise all gestures and that synchrony between behaviours is achieved by assigning sync-points of one behaviour to a sync-points of another. It uses six sync-points, start, ready, stroke-start, stroke, stroke-end, relax and end.

This model is applicable to the project because the sentiments of the assistant are independent of the concrete animation chosen. It permits to separate the sentiment specification to the expression and it is useful to give developers the freedom to chose their own animations without changing the system functioning.

3.5.2 Specification

The assistant is able to express feelings determined by the conversation. The REST API analyses the input data and sends to the app the sentiment analysed and the assistant response together. The sentiment analysis response consists of two fields, the *score* and the *magnitude*. The score takes values from -1.0 to 1.0 and indicates how negative (-) or positive (+) the text is. The value is proportional to the length of the intervention, that is, if there is an intervention, the proportional part of the text with the sentiment found. The magnitude expresses the intensity of the emotion, from 0 to $+\infty$ [18].

These numerical values are translated to sentiment names, and the gestures associated to each sentiment name is customizable, as in BML. In this project, the unique sync-point is the start.

There is an evident limitation, the inability to distinguish different good feelings and different bad feelings, for example, the result *Score: -0.7; Magnitude: 13* express a bad feeling with relatively high intensity, but it can be sadness or hate. The animations of the expression of feelings are as generic as possible in order to alleviate this limitation.

This is an example of the JSON that the REST API sends to the clients. From this data, the app obtains the field *sentimentName* and starts the animation associated with that sentiment in the appropriate moment.

```
{
  "dataResponse": "Congratulations!",
  "emotion": {
    "score": 0.5 ,
    "magnitude": 100 ,
    "sentimentName": "euphoria"
  }
}
```


Case study

4.1 Introduction

In this chapter some selected use cases are described. This chapter pretends to explain the app functionalities and how to use it. It has to be considered that app functioning is based on audio, so it is difficult to represent the user experience in a text, but some pictures of relevant interactions are represented. First of all, the application icon is shown.

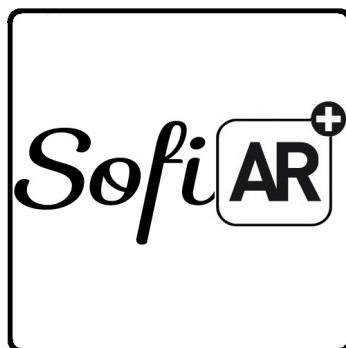


Figure 4.1: App logo

4.2 User interface

The user interface of the app is extremely simple. It consists of a grey button with a microphone symbol over the main camera capture, as can be seen in the following images. When the button is pressed its colour changes between orange and green to indicate that the app is recording audio. It is also indicated with a sound. When it stops recording the button turns grey again. In the first image, it is shown the standby state (when the app is opened or the conversation has finished) and in the second one, when it is recording.

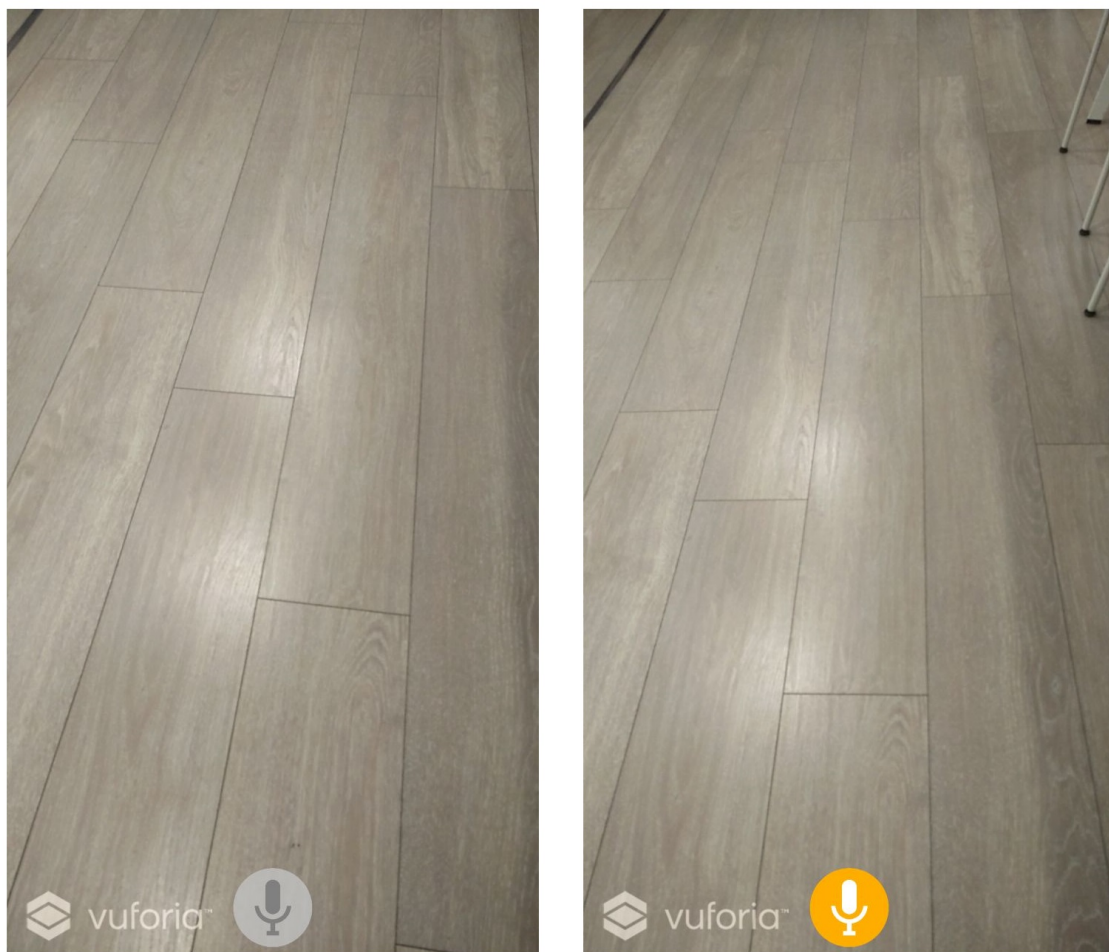


Figure 4.2: User interface

In the table of the chapter 3, which summarizes the features and differences of the AR SDKs, it is specified that the free plan of Vuforia includes a watermark but it can be removed using the enterprise edition.

4.3 Character representation

Once the app is running, as it is seen in the images above, the Vuforia engine needs the specified target to place the virtual content in the space. The used target is optimal because it has many geometrical figures and it is black and white, so the contrast is high. This target permits that most devices work well with this application because if the chosen target was any horizontal plane, only the most powerful devices could run the application well.

In the following images, it is shown how the application places the character once it has detected the target. There are two images to appreciate that it is an animated character. For example, in those two images, the character has the weight in different legs.

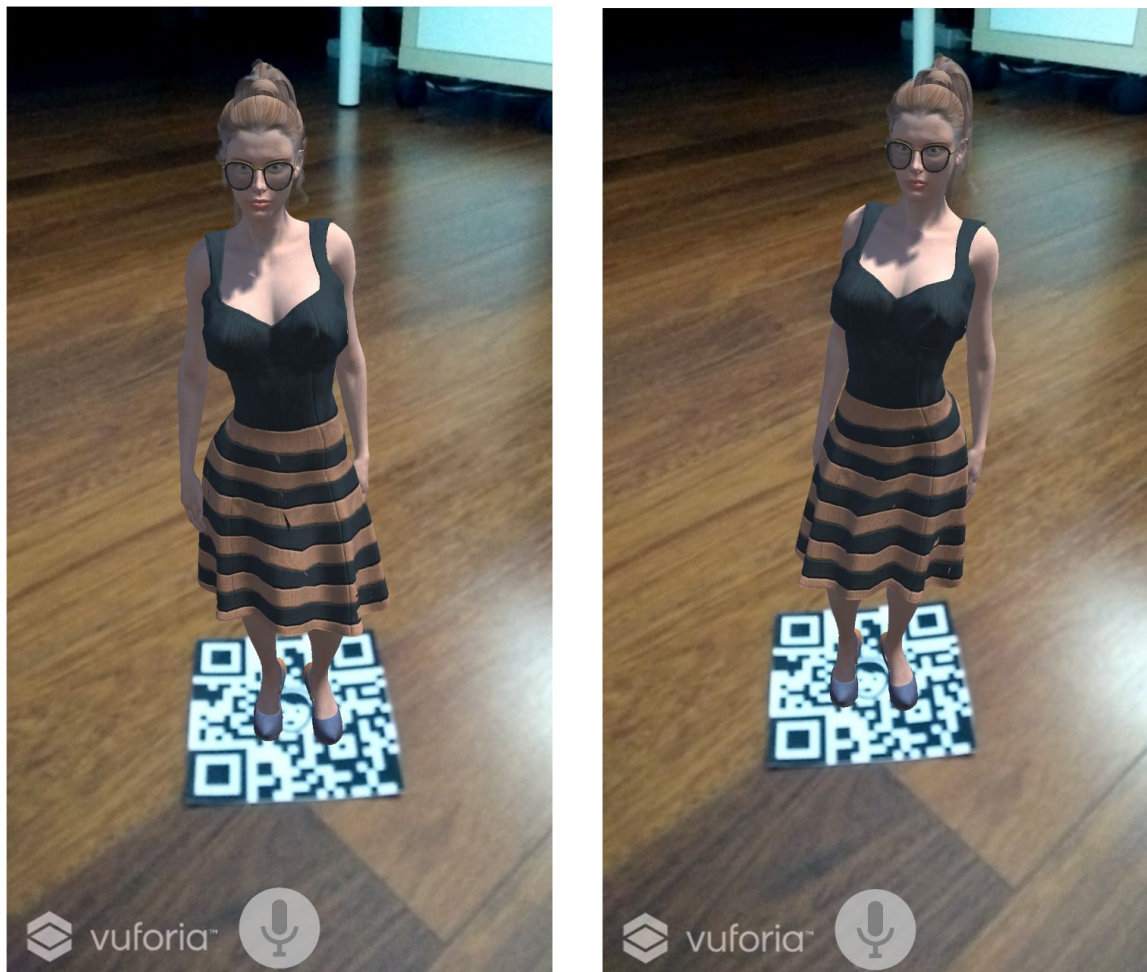


Figure 4.3: Character representation

4.4 Question-answer interaction

This section actually explains a use case, an interaction between the user and the assistant. Concretely, the user input audio is *“Hello”*. The assistant answer changes from one time to another but in the image, the answer is *“Hello, welcome! How can I help you?”*.

The animations of the assistant can be generic or can be specifically created for that answer, that is, there are some answers with a fixed sequence of movements created only for that answer, for instance, in the image. However, the capacity to create animations is less than the capacity to create peers question-answer. For this reason, there are other answers that use generic movements, like arms and head movements.



Figure 4.4: Welcome intent interaction

When the animation is not specifically created for an answer, it is needed to guarantee

that all movements are free-flowing, for example, the transition from the animation *Explain* to the animation *Idle* cannot be abrupt, if the arms are up, in the next frame the arms should not be down, they have to go down making the most natural movements possible. This free-flowing is achieved configuring the options in the Animator Asset in Unity.

4.4.1 Perceived delay

In the API architecture section, it was studied the impact of the delay in a conversation in the person, and it was compared with the delay of the API developed. Now, following the case of the image, it is shown how the perceived delay is. The next image shows the wave of the audio recorded during the interaction described.

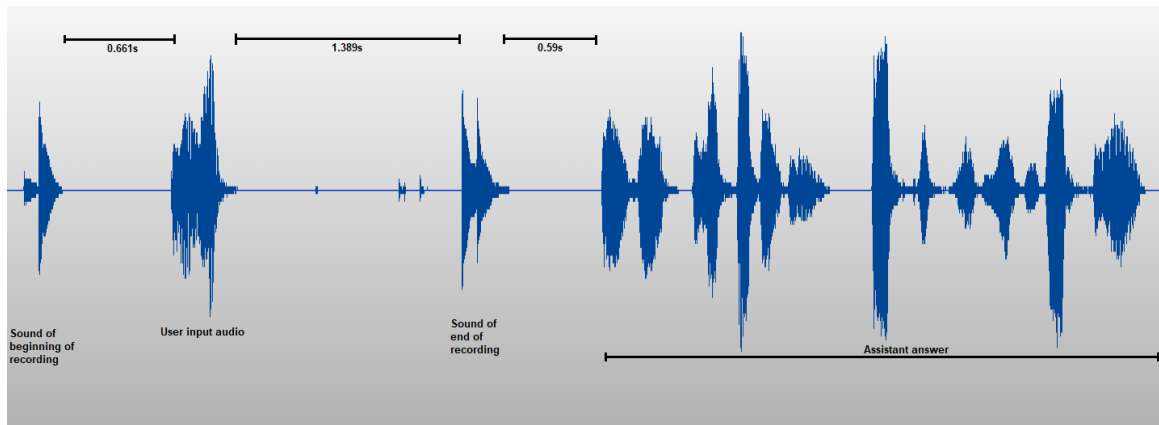


Figure 4.5: Delay in the conversation

The wave is divided into four parts that correspond with the four different sounds, the beginning of the recording, the user input data, the end of the recording and the assistant answer. Between two consecutive sounds, there is a silence that has a different duration depending on the point of the process, as it is described above.

- **After beginning of recording sound:** It is the silence since the application indicates the beginning of the recording until the user begins to speak. This silence lasts 0.661s.
- **After user input sound:** It is the silence since the user finishes to speak until the app detects that the intervention has finished and emits the sound to indicate the end of the recording. This silence lasts 1.389s.
- **After end of recording sound:** It is the silence since the application indicates the end of the recording until the assistant begins to answer. In the image, this silence

lasts 0.59s. It has to be mentioned that in the section 3.4.3.5 it was considered a minimum API delay of 749.36ms and it is more than the perceived one analysing the audio wave above. However, the app sends the request when it has decided that the intervention has finished, not when it has finished emitting the sound of the end of the recording. This is important because it is a way to create a better impression on the user because the perceived delay was only 590ms when it was more than 700ms.

4.5 Sentiments expression

It is shown the expression of two different sentiments in the character from the analysis of two conversations.

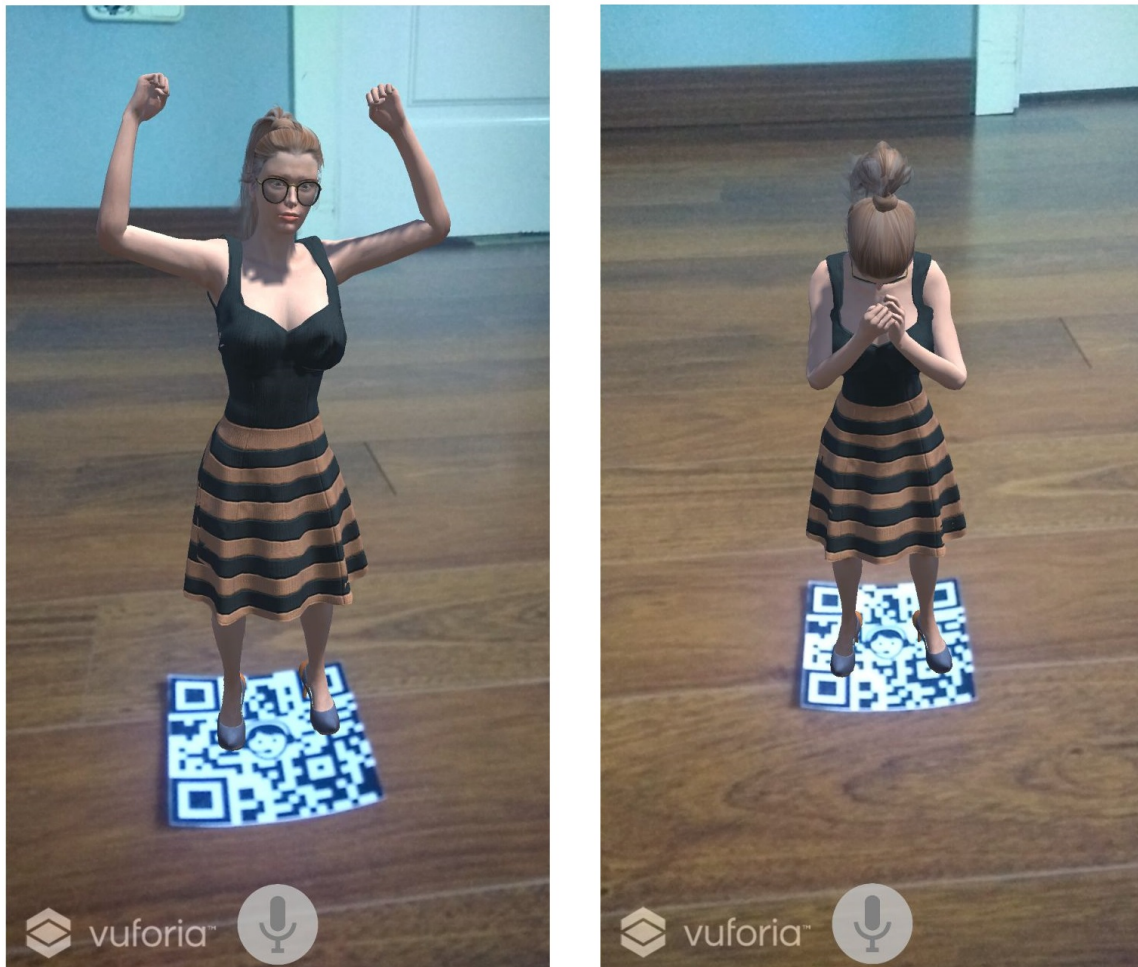


Figure 4.6: Sentiment expression

In the image on the left there is an expression of happiness and in the right of sadness. Both are representation of an intense feeling and they have been selected because they are more illustrative in a single image. However there are more animations associated with different intensities, as was explained in the sentiment analysis and generation specification.

Conclusions and future work

In this chapter, the conclusions extracted from this project are described, and also the thoughts about future work, but firstly the main problems faced during the development are explained and how they were solved.

5.1 Problems faced

5.1.1 The use of Dialogflow

The main problem faced during the project development has been how to use Dialogflow using exclusively Unity to develop the Android app. The app needed to make requests to Dialogflow API, so the first expectation was a Unity official package for Dialogflow. However, Unity only has a package for the first version of Dialogflow (API.AI) that is going to be deprecated on 23rd October 2019 and does not include some features.

The problem was that the first version used an authentication method based on tokens, so the authentication process consisted of introducing the token given by Google into the HTTP request. However, the second one introduces an authentication method based on service accounts that forces to use the client library of Dialogflow, but the library needed is

only available for Node.js, Python and Java. For this reason, it was decided to create the REST API to communicate the application and Dialogflow, delegating the authentication with Dialogflow to the REST API made with Node.js.

5.1.2 API delay

Once the creation of an intermediate API was decided, it was tried to use the Dialogflow functionality that allows sending requests directly with the user input audio. This functionality reduced the application complexity considerably, however it entailed a new problem, the delay. The process was the following.

The app recorded the audio and converted it into a group of bytes. That group of bytes was sent as a text to the intermediate API, where it was converted to audio and the request was built and sent. This method gave delays from 9s to 21s, nonsensical values for the functionality of the application. The speech to text and text to speech conversions were introduced to solve this problem. Now, only the app manipulates the audio, converting it into text and vice versa. One disadvantage of this solution is that the voice synthesizer used cannot be the Google one, that is more sophisticated than the one of the Unity package used.

As been mentioned in chapter 3, this method shows why HTTP verbs are not equivalent to the CRUD pattern. One the user input audio was sent, the verb GET did not work because it did not admit so much information, so it was needed to use POST and the goal was not to save something new in the server.

5.1.3 Avatars in Unity

It was very difficult to find good free avatars in the Unity asset store. An avatar is a character that is similar to a human. Avatars are very important because they have a particular advantage, they can exchange their animations. For this reason, before using the Reallusion software, it was tried to use all free avatars in Unity store but they have no quality. Before finding the products of Reallusion it was intended to purchase an avatar in the asset store.

5.2 Conclusions

The main project's goals have been successfully implemented. The result of the project is a functional system to offer an augmented reality virtual assistant in an Android application with a user-friendly interface, based on oral communication. The assistant is able to answer coherently to questions about the GSI smart office and when it does not know the answer. It implements a client-server architecture:

- **Server:** It is a REST API created to communicate the clients with Dialogflow and it is developed using Node.
- **Client:** The Android apps are the clients that making requests to the server control the character representation, the communication with the server and the interaction with the user.

This work shows a specific situation of the evolution of human and machine interaction, and it is expected that the future of this evolution is closely linked to this project. It is possible that instead of a smartphone, in the future we will have the assistants in smart glasses or maybe in smart lenses, but we will always need assistants. Virtual assistants in augmented reality are the future, immediate and not immediate, because despite the advance of technology, the integration of the real and virtual worlds into one will never disappear, and this is augmented reality.

5.3 Achieved goals

This section describes the achieved goals, comparing them with the initial project goals in section 1.2.

- **Develop an Android app compatible with most devices.** The app is compatible with Android devices using version 4.4 or higher, 96.2% of Android devices or 68.22% over the total.
- **The app uses augmented reality to represent an animated character.** The representation of the character is achieved using the Vuforia SDK and a 3D model created using Reallusion 3D design software.
- **The app simulate user-character communication.** The client of the system (the app) coordinates dialogue functions to simulate a real conversation. It takes the

user input data and sends it to the server. It coordinates the response and character animations.

- **The character is able to express emotions.** The app is able to express good and bad sentiments in different degrees. The sentiments expressed are determined by the NLP engine.
- **Communication is oral.** Oral communication has been achieved through speech to text and text to speech conversions because the first attempt using only audio data failed.
- **The project must help future NLP developments with Unity.** This project shows a relatively simple way to develop apps using NLP functions only with Unity. As described in future work, the API developed can be updated to admit custom Dialogflow agents.

5.4 Future work

This project entails a way to develop NLP apps with Unity, and the greater part of the future work consists of the improvement of this functionality to future developers, but also the assistant developed admits important advances.

- **Give the API the capacity to use a custom Dialogflow agent.** Now, the REST API developed only works with the Dialogflow agent created to talk about the GSI smart office. However, it is possible that the API uses a non-predefined agent, specified by the application. This improvement offers a great value for future developers that do not have to develop their own API.
- **Increase the specifically created animations.** As was mentioned, there are some responses with specifically created animation and others with generic animation and, even though the generic animations are reasonably well implemented, custom animations are more realistic.
- **Customisable character.** Now the app only admits one character, but in the future, the app can offer a list of selectable characters. If a man is implemented, voice synthesizer must change to simulate a male voice.
- **Customisable voice.** The application can be improved to give the user the ability to choose the voice of the assistant, offering several for each genre.

Impact of this project

This appendix describes quantitatively and qualitatively the possible impact of this project in different areas.

A.1 Social impact

This project develops a virtual assistant for the GSI smart office. GSI is formed by more than fifty people and all of them have access to the smart office. The virtual assistant helps them to solve their doubts about the office functioning, for instance, how to access to the office with the electronic id.

The API REST can be used for other people to develop new projects linked with the GSI smart office.

As has been mentioned, the REST API of the project entails a new way to develop applications for mobile devices with augmented reality and natural language processing, specifically for developments with Unity.

This project can be used with a new REST API to adapt it to a virtual assistant for another topic.

A.2 Ethical Impact

As has been repeated throughout the document, the interaction between the human and the machine is more and more complete. This has lead to some ethical problems that are mostly related to the context of the project than with the specific project. These problems alert that the improvement of the experience in the interactions with machines, in some cases, has become the preference to interact with virtual devices than with humans [19].

However, it is assumed that this problem is an education problem and it cannot be a reason to stop the development of the technologies that improve the experience of communicating with the machine.

A.3 Economic impact

This project develops a completely free product for GSI people, but it can be adapted to another context, generating an economic impact. For instance, it can be sold to shopping centres or to touristic places.

Generally, virtual assistants reduce the cost of the customer service for the companies, and in this case, this reduction can increase. If the customers use a more natural communication when they talk to virtual assistants, they will feel more comfortable and therefore more satisfied. If virtual assistants are better, less people working is needed.

A.4 Environmental Impact

The direct environmental impact of this project is the energy consumption of the server because the clients do not increase significantly the energy or material consumption. This is the relevant long-term impact but the current impact of the execution of the project is the energy consumed, as described below.

The development of this project, that can be considered around six hundred hours, has an energy consumption around 120KWh, taking an average for computer power (200W).

Economic budget

This appendix details an adequate budget to bring about the project proposed covering main aspects such as physical resources, human resources, licenses and taxes.

B.1 Physical resources

This project uses an Android application and a REST API. The REST API has to be deployed on a server, but in this case, it has been implemented (and it works well) on a free platform, Heroku. If it did not work correctly, it would be necessary to pay a server.

The app is executed on Android devices, so it is needed a device running Android 4.4 or later. The price of a suitable device for the execution of the application is around two hundred euros.

To develop the project a computer with at least 8 gigabytes of RAM and an Intel i5 processor or equivalent is needed. The price of this computer we can place it around one thousand euros including peripherals.

Total: 1,200€

B.2 Human resources

In this section, it is quantified the cost of development in relation to human costs. In this project that cost is only the salary of the developer.

It is considered that six hundred hours are a good approximation to the duration of development, and it is also considered that the development lasts 6 months. If five euros per hour is a standard salary of an internship, the cost of the salary is three thousand euros. However, in Spain the company has to pay thirty-five monthly in concept of social security (Seguridad Social) for an intern.

Total: 3,210€

B.3 Licenses

Reallusion software is not free, iClone costs one hundred and ninety-nine dollars and it includes Character Creator and 3D exchange costs four hundred and ninety-nine dollars. However it has been used the free trial, but it has forced to use these programs only one month.

Dialogflow, Unity, Heroku and Vuforia are free but they have plans that are not free. For instance, in Vuforia the watermark can be quit.

Total: 698\$

B.4 Taxes

The project does not have taxes associated, however in the hypothetical scenario of selling the product to a national company, it should be pay 15% of the purchase value, according with the Spanish law.

Bibliography

- [1] Dialogflow. <https://dialogflow.com/docs>. (Accessed on 02/02/2019).
- [2] Mohamed El-Zayat. Augmented reality platform for enhancing integration of virtual objects. *Cescg. org*, 2011.
- [3] Distribution dashboard — android developers. <https://developer.android.com/about/dashboards>. (Accessed on 06/19/2019).
- [4] Hannes Vilhjálmsson, Nathan Cantelmo, Justine Cassell, Nicolas Chafai, Michael Kipp, Stefan Kopp, Maurizio Mancini, Stacy Marsella, Andrew N. Marshall, Catherine Pelachaud, Zsófia Ruttkay, Kristinn Thórisson, Herwin van Welbergen, and Rick J. van der Werf. The behavior markup language: Recent developments and challenges. volume 4722, pages 99–111, 09 2007.
- [5] Chris Weider, Richard Kennewick, Mike Kennewick, Philippe Di Cristo, Robert A Kennewick, Samuel Menaker, and Lynn Elise Armstrong. Mobile systems and methods of supporting natural language human-machine interactions, May 24 2011. US Patent 7,949,529.
- [6] James H Martin and Daniel Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River, 2009.
- [7] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.
- [8] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–293. International Society for Optics and Photonics, 1995.
- [9] Martin Lechner and Markus Tripp. Arml—an augmented reality standard. *coordinates*, 13(47.797222):432–440, 2010.
- [10] Ronald Azuma. Tracking requirements for augmented reality. *Communications of the ACM*, 36(7):50–52, 1993.
- [11] Thomas Reicher, Asa Mac Williams, Bernd Brugge, and Gudrun Klinker. Results of a study on software architectures for augmented reality systems. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pages 274–275. IEEE, 2003.
- [12] Emiliano Marini. El modelo cliente/servidor. *Recuperado el*, 5, 2012.

- [13] Mark Masse. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. " O'Reilly Media, Inc.", 2011.
- [14] Jonathan Linowes and Krystian Babilinski. *Augmented Reality for Developers: Build Practical Augmented Reality Applications with Unity, ARCore, ARKit, and Vuforia*. Packt Publishing Ltd, 2017.
- [15] Sung Lae Kim, Hae Jung Suk, Jeong Hwa Kang, Jun Mo Jung, Teemu H Laine, and Joonas Westlin. Using unity 3d to facilitate mobile augmented reality game development. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 21–26. IEEE, 2014.
- [16] Authentication overview — authentication — google cloud. <https://cloud.google.com/docs/authentication/>. (Accessed on 04/16/2019).
- [17] Robert B Miller. Response time in man-computer conversational transactions. In *AFIPS Fall Joint Computing Conference (1)*, pages 267–277, 1968.
- [18] Natural language api basics — cloud natural language api — google cloud. https://cloud.google.com/natural-language/docs/basics#interpreting_sentiment_analysis_values. (Accessed on 06/01/2019).
- [19] Laurel Riek and Don Howard. A code of ethics for the human-robot interaction profession. *Proceedings of We Robot*, 2014.