**CAPITULO 2-MARCO TEORICO.**

**Definiciones y abreviaturas**

A lo largo de este proyecto aparecen una serie de términos que son utilizados con un significado concreto. A continuación se muestra una lista con las definiciones y abreviaturas utilizadas:

Realidad Aumentada (RA): Tecnología que combina técnicas de reconocimiento de formas y visualización 3D para añadir virtualidad a una imagen real de forma coherente y en función de la localización de la escena real.

Dispositivo móvil: Aparato electrónico, generalmente de tamaño pequeño, con capacidades de procesamiento, conexión de red, memoria limitada y autonomía eléctrica, diseñados específicamente para una función y que puede ser manejado con dos manos. Los más comunes son los teléfonos móviles y tabletas.

Smartphone: Teléfono inteligente. Teléfono móvil con mayores prestaciones de procesamiento y conectividad que un teléfono normal.

Tablet: Es una [computadora portátil](http://es.wikipedia.org/wiki/Computadora_port%C3%A1til) de mayor tamaño que un [teléfono inteligente](http://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente) , integrada en una [pantalla táctil](http://es.wikipedia.org/wiki/Pantalla_t%C3%A1ctil) con la que se interactúa principalmente con los dedos, sin necesidad de [teclado](http://es.wikipedia.org/wiki/Teclado_%28inform%C3%A1tica%29) físico ni [ratón](http://es.wikipedia.org/wiki/Rat%C3%B3n_%28inform%C3%A1tica%29).

Kit de desarrollo de software (SDK): Conjunto de herramientas de desarrollo de software que permite al desarrollador crear aplicaciones para un sistema concreto.

API (Interfaz de programación de aplicaciones) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Java: Es un [lenguaje de programación](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) de [propósito general](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general), [concurrente](http://es.wikipedia.org/wiki/Computaci%C3%B3n_concurrente), [orientado a objetos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) y basado en [clases](http://es.wikipedia.org/wiki/Clase_%28inform%C3%A1tica%29) que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

GPS: El [sistema global de navegación por satélite](http://es.wikipedia.org/wiki/Sistema_global_de_navegaci%C3%B3n_por_sat%C3%A9lite) permite determinar en todo el mundo la [posición](http://es.wikipedia.org/wiki/Posici%C3%B3n) de un objeto, una persona o un vehículo con una precisión hasta de centímetros, aunque lo habitual son unos pocos metros de precisión.

Marker: Marcador o imagen utilizada en algunas aplicaciones de realidad aumentada y cuya posición condiciona el posicionamiento del objeto.

Target: Marcador natural o imagen que forma parte del entorno natural de una aplicación de RA y que sustituye a los clásicos marcadores.

Head Mounted Display (HMD): Dispositivo de visualización similar a un casco, que permite reproducir imágenes creadas por ordenador sobre un "display" muy cercano a los ojos.

Realidad Virtual (RV): Es una ciencia basada en el empleo de ordenadores y otros dispositivos, cuyo fin es producir una apariencia de realidad que permita al usuario tener la sensación de estar presente en ella.

3G: Es la abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil mediante UMTS (servicio universal de telecomunicaciones móviles).

ASL: Abreviatura para Lenguaje de signo Americano.

LSE: Abreviatura para Lenguaje de signos Español.

iOS (por sus siglas en [inglés](http://es.wikipedia.org/wiki/Ingl%C3%A9s) *iPhone/iPod/iPad Operating System)*: Es un [sistema operativo](http://es.wikipedia.org/wiki/Sistema_operativo) móvil de la empresa [Apple Inc](http://es.wikipedia.org/wiki/Apple_Inc.).

Android: Es un [sistema operativo](http://es.wikipedia.org/wiki/Sistema_operativo) diseñado principalmente para dispositivos móviles con [pantalla táctil](http://es.wikipedia.org/wiki/Pantalla_t%C3%A1ctil), como [teléfonos inteligentes](http://es.wikipedia.org/wiki/Tel%C3%A9fonos_inteligentes) o [tabletas](http://es.wikipedia.org/wiki/Tableta_%28computadora%29), y también para relojes inteligentes, televisores y automóviles, inicialmente desarrollado por Android, Inc.

Middleware: Es un [software](http://es.wikipedia.org/wiki/Software) que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos.

Vuforia: Es un SDK que permite construir aplicaciones basadas en la realidad aumentada que según su propia descripción, esta utiliza la pantalla del dispositivo como un "lente mágico" en donde se entrelazan elementos del mundo real con elementos virtuales como letras, imágenes, objetos 3d, etc.

Tracker: Contiene los algoritmos de visión computacional para detectar y seguir los objetos en los fotogramas capturados por la cámara.

Render o Renderizado: es un término usado para referirse al proceso de generar una imagen o vídeo mediante el cálculo de iluminación GI partiendo de un modelo en 3D.

Patrones de diseño: Un patrón de diseño resulta ser una solución a un problema en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces e diseño. Para que una solución sea considerada un patrón debe poseer ciertas características como su efectividad resolviendo problemas similares en ocasiones anteriores. También debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Patrones arquitectónicos: también llamados arquetipos ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor.

Framework (marco de trabajo): se define, en términos generales, como un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Modelo–vista–controlador (MVC): es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones..

DAO (Objeto de Acceso a Datos): es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.

Facade (*Fachada*): es un tipo de patrón estructural que conoce qué clases del subsistema son responsables de una determinada petición, y delega esas peticiones de los clientes a los objetos apropiados del subsistema.

Hibernate: es una herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de [NHibernate](http://es.wikipedia.org/wiki/NHibernate)) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los [beans](http://es.wikipedia.org/wiki/JavaBean) de las entidades que permiten establecer estas relaciones.

Base de datos (DB o DDBB): es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Ejemplos: MySql, Oracle,ect.

API (Interfaz de programación de aplicaciones): es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

JDBC *(Java Database Connectivity)*: es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

SQL ( lenguaje de consulta estructurado): es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas

[Open source](http://es.wikipedia.org/w/index.php?title=Open_source&redirect=no) *(Código abierto):* es la expresión con la que se conoce al software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

DTD (Definición de Tipo de Documento): Descripción en donde etiquetas y atributos son usados para describir contenido (estructura y sintaxis) de documentos SGML, XML o HTML. Su función básica es describir el formato de los datos. De esta manera se mantiene un formato común y consistente en todos los documentos que utilicen la misma DTD.

Polimorfismo: se refiere a la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos. El único requisito que deben cumplir los objetos que se utilizan de manera polimórfica es saber responder al mensaje que se les envía.

Persistencia: se refiere a la propiedad de los datos para que estos sobrevivan de alguna manera. De forma sencilla puede entenderse que los datos tienen una duración efímera; desde el momento en que estos cambian de valor se considera que no hay persistencia de los mismos. Sin embargo en informática hay varios ámbitos donde se aplica y se entiende la persistencia.

Mapeo objeto-relacional ( Object-Relational mapping): es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia

**Lenguaje de Señas:**

Uno de los primeros estudios de los sistemas de signos como medio de comunicación para personas sordas fueron realizadas por Juan de Pablo Bonet ,(1573-1633) pedagogo español que escribió “Reducción de las letras y Arte para enseñar a hablar los Mudos “, en este libro se mostraba un método de comunicación mediante alfabeto de signos para las personas sordas.

A partir del alfabeto publicado por J. Bonet, Charles-Michel de l'Épée (1712-1789) publica un alfabeto en el que se basan los alfabetos de signos que se usan en la actualidad. Dicho alfabeto fue publicado por su sucesor el abate Sicard en la forma de “Diccionario general de Signos”.

En 1817 Thomas Gallaudet abrió una escuela de sordos en Connecticut (Estados Unidos) donde sentó las bases para la ASL, tomando como referencia la lengua de signos francesa y algunos signos utilizados por las tribus indias de Norteamérica. En 1965 William Stokoe, describió una gramática precisa para el lenguaje de signos americano en el libro “A Dictionary of American Sign Language on Linguistic Principles” [López, 2009 ].

La lengua de signos es un lenguaje no oral, no auditivo, sino gestual y visual que construyo un nuevo campo de investigación para la ciencia lingüística, centrada, hasta entonces, en los lenguajes hablados [[Rodríguez González](http://www.cervantesvirtual.com/obras/autor/4364/Rodr%C3%ADguez%20Gonz%C3%A1lez,%20Mar%C3%ADa%20%C3%81ngeles), 2003].

En la actualidad existen estudios de los lenguajes de signos, los más conocidos son el ASL(Lenguaje de signo Americano) de Norte América y el LSE (Lenguaje de signos Español) realizado en España ,además existen trabajos sobre los lenguajes de signos gestuales utilizados en otros países de Europa como Gran Bretaña, Francia, Suecia, Dinamarca y Holanda.

A pesar de estos estudios el Lenguaje de signos no es universal y puede variar dependiendo la región en la que nos encontremos. Por ello el sistema deberá presentar los símbolos del alfabeto LSE que son utilizados por la comunidad sorda de Jujuy.

**Que es realidad aumentada?**

De una forma vulgar se puede decir que la Realidad Aumentada es un sistema que potencia las capacidades de nuestros sentidos. Similar al trabajo que realiza una lupa, un microscopio, etc. pero, en este caso ampliando la percepción que el usuario tiene de la realidad real mediante la inclusión de elementos virtuales en la misma.

De una forma más técnica, según Ronald Azuma, la Realidad Aumentada (RA) es un entorno que incluye elementos de Realidad Virtual y elementos del mundo real. Por ejemplo, un usuario de RA puede llevar un dispositivo móvil, a través de las cuáles puede ver el mundo, así como imágenes generadas por computadora que se proyectan encima de ese mundo. Siguiendo esta definición, un sistema de RA es aquel que:

* Mejora el mundo real(agregando nueva información)
* Es interactivo en tiempo real
* Se registra en 3 dimensiones

En definitiva, podemos decir que un sistema de RA necesita un dispositivo que se encargue de recoger información sobre la realidad, un ordenador capaz de crear imágenes sintéticas y procesar la imagen real añadiendo esta información (procesador + software) y un medio que permita proyectar la imagen final (pantalla) [Rolando, 2012 ].

**Historia de la Realidad Aumentada**

A principios de los años 60’s varias personas comenzaron a pensar en algún tipo de sistema que pudiese crear un nuevo mundo diferente al real, un mundo que llamarían “realidad virtual”. En 1962, un director de fotografía, Morton Heilig crea un simulador de moto llamado ‘Sensorama’ con imágenes, sonido, vibración y olfato. Varios años más tarde, en 1966, Ivan Sutherland inventa el HMD, display de cabeza, lo que sugiere una ventana a un mundo virtual.

En el año 1975, Myron Krueger, crea “Videoplace”, un laboratorio de realidad virtual, que diseña un sistema que permite a los usuarios interactuar con objetos virtuales. Jaron Lanier fundador de “VPL Research” popularizó a mediados de los 80’s el término “Realidad Virtual”.

En el año 1992 Steven Feiner, Blair MacIntyre y doree Seligmann diseñan KARMA, un prototipo de un sistema de Realidad Aumentada. En el año 2000, los sistemas informáticos llega el “boom” de la realidad aumentada, y se presenta ARQuake, el primer juego al aire libre con dispositivos móviles de Realidad Aumentada. A finales del 2008 sale a la venta AR WIkitude Guía, una aplicación para viajes y turismo desarrollada para la plataforma Android.

En 2009 se crea el logo oficial de la Realidad Aumentada con el fin de estandarizar la identificación de la tecnología aplicada en cualquier soporte o medio por parte del público general [Rolando, 2012 ].



Figura 2.3 Logo oficial de Realidad Aumentada [Total Inmersion, 2011]

**Antecedentes Actuales de Realidad Aumentada**

Hasta ahora, la mayoría de los proyectos o productos de realidad aumentada que se han dado a conocer en ferias o que han llegado al mercado están enfocados a los videojuegos, al turismo o a la publicidad. Son pocas las empresas que desarrollaron aplicaciones destinadas a ayudar a personas con capacidades diferentes, se puede mencionar:

**“accentac” es una nueva aplicación móvil creada por Pixtorm para mejorar la movilidad en Madrid.** La aplicación accentac, de carácter gratuito, describe y sitúa los servicios necesarios para facilitar la accesibilidad y movilidad de los mismos en rampas, ascensores, baños, accesos, salidas de emergencia, puntos de información. Esta aplicación es muy útil para los ciudadanos con discapacidad visual, auditiva o con dificultades de movilidad que necesiten un apoyo al estar en un nuevo entorno[TICbeat, 2012].

En el ámbito del cine, hay que destacar que en el Instituto de Desarrollo Tecnológico y Promoción de la Innovación de la Universidad Carlos III de Madrid se ha desarrollado la aplicación ‘Whatscine’; un sistema de accesibilidad de bajo coste para cine digital que muestra audio descripciones, subtitulado y traducción al lenguaje de signos de las películas para ayudar a las personas con discapacidad auditiva y visual[Bernat, 2013].

Las compañías estadounidenses, Fundación Vodafone y Fundación Aprocor han desarrollado una serie de aplicaciones de realidad aumentada para ayudar a la integración de personas con discapacidad intelectual en el entorno laboral. El proyecto de realidad aumentada viene a ser un tutorial pensado para discapacitados intelectuales, esto se logra gracias a una cámara, una conexión 3G y un dispositivo como una tablet. Concretamente, esta aplicación ha sido desarrollada para solucionar tres tareas: cómo llegar a un sitio, identificar lugares o personas clave en el trabajo y aprender a utilizar una máquina [*Europa Pres*, 2012].

**Android**

Android es un sistema operativo móvil basado en Linux, que junto a un conjunto de aplicaciones y herramientas middleware trata de proporcionar una

plataforma de desarrollo para dispositivos como smartphones o tablets. Al igual que otros sistemas operativos móviles como iOS, Symbian o BlackBerry, fue pensado específicamente para smartphones pero sin embargo Android presenta una peculiaridad que le ha permitido crecer a un ritmo mucho mayor que sus competidores hasta llegar ser la plataforma más extendida. Esta peculiaridad es que al estar basado en Linux, resulta una plataforma libre, gratuita y multiplataforma.

Otra característica que lo hace popular entre los desarrolladores de aplicaciones es el hecho de estar pensado en un sublenguaje de Java que facilita el desarrollo de estas aplicaciones al ser un lenguaje muy extendido y conocido. Esto proporciona herramientas de acceso a componentes del teléfono o dispositivo tales como GPS o la agenda de una manera fácil e intuitiva. Además, Android proporciona una API que nos extiende la ya extensa API de Java pero más enfocada a la optimización con el propio sistema operativo [IñarreaSagüés, 2012].

**Herramientas Necesarias:**

Java JDK: El JDK, Java Developmente Kit, se trata de un conjunto de herramientas software para la creación del programa en lenguaje Java. El JDK de Java es necesario para crea aplicaciones en Android e instalación del SDK de Android, por ello necesario para el desarrollo de nuestro proyecto.

Android SDK: Software Development Kit (SDK) o paquete de desarrollo de software propio de Android para la creación de aplicaciones para el sistema operativo Android. Consta de un depurador de código, bibliotecas para la programación en el lenguaje, un simulador de teléfonos, documentación y ejemplos de código de programación. El SDK nos proporciona un gestor de versiones del propio sistema operativo con el que podemos descargar las librerías adicionales de cada versión y nos notifica si alguno de los métodos usados están obsoletos. Además, nos permite acceder y controlar dispositivos Android correctamente conectados a nuestro ordenador.

Target Management System: Se trata de la herramienta online proporcionada por la arquitectura Vuforia para la creación de los patrones de reconocimientos utilizados por el SDK Vuforia para la identificación de los objetos y su posterior tratamiento. Para ello, la realidad aumentada basada en el visionado, a través de cualquier dispositivo con cámara, debe tener lo que se conoce como “DataSet”, con el que compara la imagen captada por la cámara . El “Target Management System” ofrece una herramienta basada en la web para los desarrolladores que utilizan el SDK Vuforia para crear esos “DataSet” a partir de una imagen importada a la herramienta. El objeto que nos crea, se quedará almacenado en la página de Vuforia y podremos descargarnos para su utilización en nuestra aplicación [IñarreaSagüés, 2012].

Para acceder a la herramienta por lo tanto, primero habremos de registrarnos

en la página de Vuforia como desarrolladores del mismo.

**Realidad aumentada vs. Realidad virtual**

La realidad virtual es una tecnología que abarca un amplio espectro de ideas. El termino de Realidad Virtual fue creado por Jaron Lanier, fundador de “VPL Research”, quien la definió como “un entorno generado por una computadora, interactiva, tridimensional en el cuál se introduce a la persona”. Hay tres puntos claves en esta definición:

* Este entorno virtual está generado por una computadora mediante una escena en tres dimensiones, la cual requiere una alta capacidad de gráficos por parte del ordenador para adecuar el nivel de realismo.
* El mundo virtual es interactivo pues el usuario requiere una respuesta en tiempo real desde el sistema para poder interactuar en él de una manera efectiva.
* El usuario está inmerso en el mundo virtual.

Una diferencia importante entre los sistemas de Realidad Virtual y los sistemas de Realidad Aumentada es la inmersión de la persona en el entorno. En los sistemas de Realidad Virtual, la persona se encuentra en un mundo totalmente virtual donde el entorno está bajo control del sistema. Sin embargo, los sistemas de Realidad Aumentada se encargar de “ampliar” la escena del mundo real manteniendo en el usuario una sensación de presencia en el mundo real. Las imágenes virtuales están mezcladas con la visión del mundo real, creando una visión aumentada dando al usuario la ilusión que los objetos de los mundos real y virtual coexisten.

En los sistemas de Realidad Virtual, el usuario está completamente inmerso en un mundo artificial y no hay manera de interactuar con objetos del mundo real. En cambio, en los sistemas de Realidad Aumentada, los usuarios pueden interactuar mezclando el mundo real y virtual de una forma natural. Así, la diferencia entre Realidad Virtual y Realidad Aumentada está en el tratamiento que hacen del mundo real [Rolando, 2012 ].

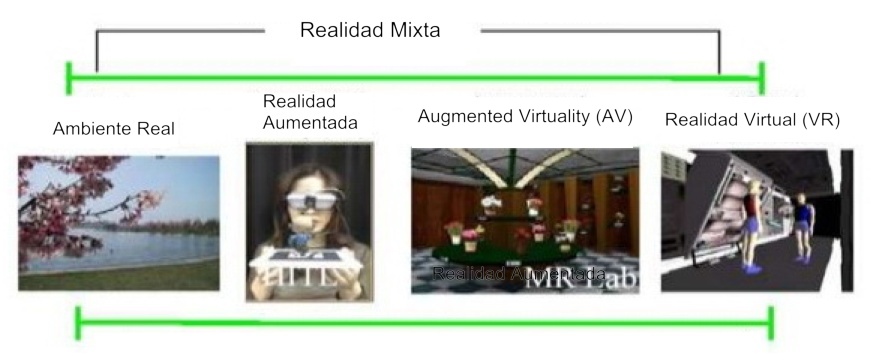


Figura 2.1 Múltiples realidad que existen en la actualidad [Rolando, 2012 ]

**Reconocimiento de marcador**

Los marcadores son hojas de papel o imágenes con símbolos que el software interpreta realizando una respuesta específica para un marcador específico.

Las aplicaciones con marcadores toman fotograma a fotograma de una cámara, bien sea de móvil o webcam, para procesarlo y localizar patrones de imagen conocidos como el mostrado más abajo. Una vez que el sistema localiza uno de los marcadores reconocibles, mezcla la imagen real con su parte virtual mostrando sobre el marcador el objeto que deseemos, tanto en tres dimensiones como en dos dimensiones [Rolando, 2012 ].

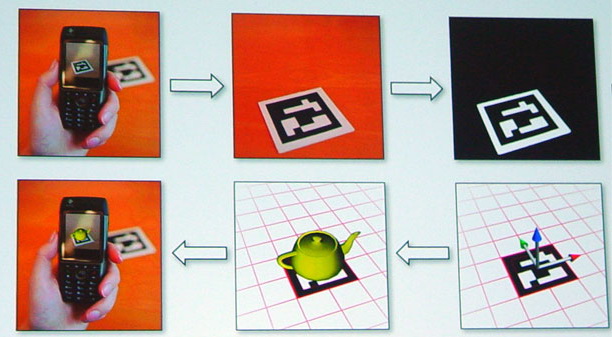
****

Figura 2.2 *Imagen que ilustra el funcionamiento del reconocimiento de marcadores [Sag,2013]*

**Realidad aumentada en dispositivos móviles**

Hoy en día los dispositivos móviles están al alcance de la mayoría de las personas, estos cuentan con las prestaciones necesarias para servir como soporte de estas aplicaciones y sacar el máximo provecho de ellas. La mayor parte de los dispositivos móviles adquiridos por los usuarios disponen de cámara de vídeo de alta resolución y alta capacidad de procesado. Este hecho, y los recientes estudios de mercado, hacen pensar que la integración de la RA en el creciente mercado de los dispositivos móviles forman un campo de investigación de interés [Clemens, 2011].

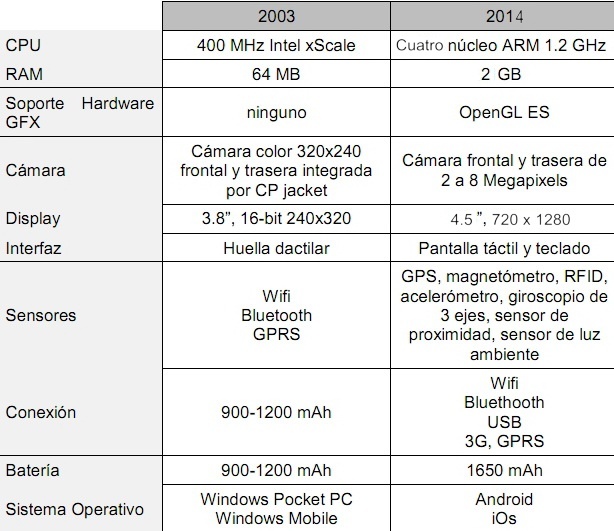


Figura 2.4 Comparación del avance tecnológico en los dispositivos móviles[Clemens, 2011].

En la Tabla comparativa se observa cómo han mejorado los dispositivos móviles en los aspectos más relevantes para RA: procesadores más rápidos, más memoria, mejores interfaces de entrada, pantallas más grandes y de mayor calidad gráfica, más sensores y mejora de las posibilidades de conexión. La tabla comparativa presentada es una actualización de la presentada por Clements & Schmalstieg.

Con la llegada de los sistemas operativos iOS y Android, el término Smartphone se acuñó para indicar la capacidad de estos dispositivos para la implementación de aplicaciones informáticas complejas. La mejora de interfaces, y en particular la introducción de la pantalla táctil, y el fácil acceso a las aplicaciones para los usuarios, explica su gran éxito comercial. Este fuerte interés comercial también ha traído consigo la mejora de las herramientas de desarrollo de software dedicado a smartphones.

Gracias a los avances tecnológicos, los dispositivos móviles están cada vez mejor preparados para ejecutar las aplicaciones de RA cada vez más complejas.

**Tecnologías de Realidad aumentada**

En la actualidad existen diferentes tecnologías relacionadas a la implementación de un Prototipo de realidad aumentada para dispositivos móviles. Algunas tecnologías que podemos mencionar son:

* **Unity3D:** Es un motor gráfico 3D para PC y Mac que viene empaquetado como una herramienta para crear juegos, aplicaciones interactivas, visualizaciones y animaciones en 3D.
* **Vuforia:** Es un frameworks para el desarrollo de aplicaciones de realidad aumentada para Android e iOS. Se basa en el reconocimiento de marcas naturales incluyendo objetos 3D[Mamolar, 2012].
* **Metaio:** Es una plataforma de desarrollo de aplicaciones de realidad aumentada para dispositivos Android e iOS. Las aplicaciones se basan en el Reconocimiento de marcas naturales, e integra la gravedad en los módulos de reconocimiento para añadir precisión [Mamolar, 2012].
* **AndAR:** Es una librería creada en el 2010 para dispositivos Android y utiliza una API escrita en java [Mamolar, 2012].
* **NyARToolkit:** Es un SDK de código abierto para el desarrollo de aplicaciones de realidad aumentad basadas en el reconocimiento de marcadores[Mamolar, 2012].

**Librería Vuforia**

Según el análisis realizado en el capítulo 3, se eligió como tecnología del proyecto a Vuforia. Su funcionamiento se basa en la detección de ciertas imágenes usando la cámara del dispositivo (**Trackables**). Las funciones de la librería proporcionan la posición y orientación de dichos trakeables a través de una matriz de 4×4 llamada **matriz de pose.**

**Esta** plataforma de desarrollo de software pone a disposición de los programadores de aplicaciones móviles un motor de reconocimiento de imágenes muy potente, así como un amplio abanico de herramientas diseñado para permitirles dar rienda suelta a su creatividad sin que se vean obligados a preocuparse por las limitaciones de índole técnica.



Figura 2.5 Arquitectura de la aplicación estándar [Mamolar, 2012].

El desarrollador debe disponer previamente de las imágenes que quiere reconocer con su aplicación. Las cuales deben ser subidas a la Web Management System de Vuforia y, o bien se crea una base de datos de imágenes alojada en la nube que se consultará en tiempo de ejecución, o bien se descarga dicha base de datos y se incluye en la aplicación. Durante la ejecución de la aplicación, esta interacciona con el motor de Vuforia para reconocer las imágenes almacenadas en la base de datos. Cuando el motor de Vuforia detecta una imagen en la escena, se puede proyectar el contenido virtual creado por el desarrollador.

OpenGL

Es una librería gráfica escrita originalmente en C que permite la manipulación de gráficos 3D a todos los niveles. Esta librería se concibió para programar en maquinas nativas Silicon Graphics bajo el nombre de GL (Graphics Library). Posteriormente se considero la posibilidad de extenderla a cualquier tipo de plataforma y asegurar así su portabilidad y extensibilidad de uso con lo que se llego al termino Open Graphics Library, es decir, OpenGL.

La librería se ejecuta a la par con nuestro programa independientemente de la capacidad gráfica de la maquina que usamos. Así esta librería puede usarse bajo todo tipo de sistemas operativos e incluso usando una gran variedad de lenguajes de programación. No obstante, su uso más extenso suele ser el lenguaje C o C++. [García Guevara, 2004 ].

Se divide en tres partes funcionales:

* La librería OpenGL, que proporciona todo lo necesario para acceder a las funciones de dibujado de OpenGL.
* La librería GLU (OpenGL Utility Library), una librería de utilidades que proporciona acceso rápido a algunas de las funciones más comunes de OpenGL., a través de la ejecución de comandos de más bajo nivel, pertenecientes a la librería OpenGL propiamente dicha [REDBOOK].
* GLX (OpenGL Extension to the X Window System) proporciona un acceso a OpenGL para poder interactuar con un sistema de ventanas X Window, y está incluido en la propia implementación de OpenGL (su equivalente en Windows es la librería WGL, externa a la implementación de OpenGL).

Además de estas tres librerías, la librería GLUT (OpenGL Utility Toolkit) proporciona una interfaz independiente de plataforma para crear aplicaciones de ventanas totalmente portables [GLUT] [García, 2003 ].

La librería OpenGL viene integrada en Unity3D y es la encargada del renderizado de los objetos 3D, manipular matrices de transformación activa y de otras tareas graficas en bajo nivel.

**Arquitectura para Realidad Aumentada**

Una aplicación de Realidad Aumentada basada en Vuforia, consta de una serie de componentes muchos de estos son singletons. Un singleton consiste en una clase diseñada para tener sólo una instancia (o un número limitado de ellas), por comodidad a la hora de emplear el lenguaje, se llamará instancia a un singleton.

• Cámara (camera): La instancia de la cámara se encarga de que cada fotograma capturado por la cámara digital se pase de forma eficiente al tracker.

• Convertidor de imágenes (Image Converter): La instancia del conversor de formato de pixel realiza la conversión entre el formato con el que trabaja la cámara a un formato adecuado para el renderizado en OpenGL ES y para el seguimiento (por ejemplo, luminancia).

• Tracker: El tracker contiene los algoritmos de visión computacional para detectar y seguir (detect & track) los objetos en los fotogramas capturados por la cámara. Basado en la imagen tomada por cámara, diferentes algoritmos se ocupan de detectar nuevas imágenes de referencia (Targets) o marcadores (Markers), y evaluar los botones virtuales (Virtual Buttons). Los resultados se almacenan en un objeto de estado que es utilizado por el procesador de vídeo de fondo y al que puede accederse desde el código de la aplicación. El tracker puede cargar múltiples conjuntos de datos (datasets), pero sólo uno puede estar activo a la vez.

• Procesador de vídeo de fondo (Video Background Rendered, singleton): La instancia del procesador de vídeo de fondo procesa la imagen capturada por la cámara que se encuentra almacenada en el objeto de estado.

• Código de la aplicación (Application Code): El desarrollador ha de inicializar todos los componentes anteriores y llevar a cabo tres pasos fundamentales en el código de la aplicación. Por cada fotograma procesado, el objeto de estado se actualiza y se llama al método de procesamiento de la aplicación. El desarrollador debe:

* Consultar el objeto de estado para los Targets y/o Markers nuevos que puedan aparecer en escena o si se actualiza su estado.
* Actualizar la lógica de la aplicación con nuevos datos de entrada.
* Renderizar la capa de Realidad Aumentada.

• Recursos de imágenes de referencia (Target Resources): Los Target Resources se crean mediante el Target Management System, que se encuentra disponible online. El dataset descargado contiene un fichero de configuración XML que permite al desarrollador configurar ciertas características de los trackables y un fichero binario que contiene la base de datos de los trackables. Estos elementos (XML y binario) son compilados por la aplicación a desarrollar en el paquete de instalación de la aplicación y los usa el SDK Vuforia en tiempo de ejecución.



Figura 2.6 Diagrama de flujo de datos en una aplicación

de ejemplo [Mamolar, 2012].

La figura muestra el diagrama de flujo de datos en una aplicación de ejemplo que usa el SDK de Vuforia. La cámara del dispositivo obtiene un frame que se convierte al formato requerido y se pasa al tracker, que puede tener distintas configuraciones. Este proyecto se ha basado en el tracker ImageTargets, es decir, reconoce un solo target o marcador en el frame. Esto es así porque la superficie plana se considera como un marcador-imagen, y suponemos que en la escena solo vamos a tener una única superficie plana dominante, por tanto no tendrá sentido poder reconocer dos marcadores, es decir, dos superficies planas distintas sobre las que proyectar nuestro modelo. El tracker obtiene la lista de marcadores que tiene que buscar de la base de datos alojada en la nube, o integrada en la aplicación, o se han podido crear los marcadores en tiempo de ejecución a partir de frames obtenidos por la cámara del dispositivo móvil. Esta última opción es la que se ha desarrollado en el proyecto. Finalmente, el tracker indica dónde debe renderizarse el modelo, y el nuevo frame se proyecta en la pantalla.

Procesos de diseño 3D

El modelado, consiste en construir objetos a partir de elementos geométricos. Los objetos 3D son sólidos virtuales formados por datos. Los datos tridimensionales se presentan en pantalla mediante líneas, puntos, degradados de color, imágenes o una combinación de todos ellos. Tradicionalmente, se ha modelado siempre en modo línea o en modo puntos. Sin embargo, desde hace unos años la potencia de los ordenadores y el uso de las capacidades 3D en tiempo real, ha permitido la posibilidad de trabajar en modo casi-fotográfico. Esta técnica se conoce normalmente como Open-GL. A pesar del atractivo indudable del modelado en Open-GL, muchas veces el modo línea puede ser la forma más efectiva de afrontar un proyecto. A menudo, se alterna de modo de visualización, a lo largo del proceso de modelado de los objetos.

El texturado, consiste en recubrir nuestros objetos, o mejor dicho la superficie de nuestros objetos, con imágenes que le proporcionen un aspecto lo más realista posible. Las texturas, pueden ser fotográficas o procedurales. Las primeras, son fotografías digitalizadas. Las segundas, son producidas mediante algoritmos matemáticos. También se pueden aplicar a los objetos, superficies rugosas o suaves para imitar esta propiedad de los objetos reales. Aprender a texturar, es básico para obtener un render mínimamente realista. Es necesario un trabajo importante de retoque y redimensionado de imágenes (mediante un programa como Gimp), antes de ponerse manos a la obra con Blender. Un proyecto concienzudo y una preparación meticulosa de las imágenes, nos evitará muchos quebraderos de cabeza. Los programas de 3D, son siempre difíciles, así que una buena planificación del trabajo permite utilizar las técnicas de 3D más sencillas y eficaces[Martinez, 2006].

Blender

Blender es un programa que integra una serie de herramientas para la creación de un amplio rango de contenidos 3D, con los beneficios añadidos de ser multiplataforma y tener un tamaño de unos 5MB.

Destinado a artistas y profesionales de multimedia, Blender puede ser usado para crear visualizaciones 3D, tanto imágenes estáticas como vídeos de alta calidad, mientras que la incorporación de un motor de 3D en tiempo real permite la creación de contenido interactivo que puede ser reproducido independientemente.

Originalmente desarrollado por la compañía 'Not a Number' (NaN), Blender es ahora desarrollado como “Software Libre”, con el código fuente disponible bajo la licencia GNU GPL [Roosendaal, 2004].

Características principales:

* Paquete de creación totalmente integrado, ofreciendo un amplio rango de herramientas esenciales para la creación de contenido 3D, incluyendo modelado, mapeado uv, texturizado, rigging, weighting, animación, simulación de partículas y otros, scripting, renderizado, composición, post-producción y creación de juegos.
* Multiplataforma, con una interfaz unificada para todas las plataformas basada en OpenGL, listo para ser usado en todas las versiones de Windows, Linux, OSX, FreeBSD, Irix y Sun, y otros sistemas operativos.
* Arquitectura 3D de alta calidad permitiendo un rápido y eficiente desarrollo.
* Canales de soporte gratuito vía http://www.blender3d.org .
* Una comunidad mundial de más de 250.000 usuarios.
* Tamaño pequeño de ejecutable para una fácil distribución
* Es un software libre.

Blender se convierte en la herramienta ideal para realizar modelado de objetos 3d necesarios para la aplicación.

**Motores Gráficos**

Un motor de grafico es un conjunto de herramientas que realizan cálculos geométricos y físicos utilizados en aplicaciones y videojuegos. Este conjunto de utilidades representa un simulador ágil en tiempo real que reproduce las características de las aplicaciones y videojuegos.

Puesto que las funcionalidades proporcionadas por un motor son limitada. La manipulación del motor se hace generalmente a través de un lenguaje de script o por la interfaz.

Algunas de los componentes que forman el motor pueden ser la carga, la animación de modelos, la detección de colisiones entres objetos, la física, interfaces gráficas de usuario o incluso herramientas de inteligencia artificial. Mientras que el contenido del una aplicación está compuesto por los modelos, texturas específicas, el comportamiento ante las colisiones de objetos o la forma que tienen los objetos de interactuar con el entorno.

Un motor de gráfico debe ofrece como mínimo las siguientes utilidades:

* Un motor 3D que permite la creación y visualización de un universo 3D.
* Un motor audio que permite la integración de elementos sonoros y música.
* Un motor físico que permita gestionar los comportamientos físicos de los objetos en un universo 3D.
* Herramientas de gestión de red.

Existen diferentes maneras de clasificar los motores de gráficos dependiendo de las características que nos interesan. Una de estas características es el coste, efectivamente existen motores gratuitos o de pago que pueden llegar hasta el medio millón de dólares. Estos últimos son destinados a estudios de desarrollo profesional con gran poder financiero. Estos motores ofrecen un marco de aplicación y en algunos casos un entorno de desarrollo integrado. Este entorno permite crear o importar todos los elementos que se quieren integrar en la aplicación y determinar mecánicas de la aplicación con la ayuda de los scripts.

Los motores gratuitos están destinados a los desarrolladores principiantes o independientes y no suelen tener interfaz global. Para utilizar estos motores hay que descargar un SDK , un kit de desarrollo que permite utilizarlos, pero se debe generar el programa entero a mano, escribiendo código: Crystal Space, Ogre 3D, Irrlicht.

Los motores intermediarios, en esta categoría se incluyen los motores con herramientas de desarrollo más potentes. Estos motores disponen de un entorno de desarrollo integrado (IDE) que permite la creación de programas manipulando los elementos en una interfaz gráfica y generando automáticamente una gran parte del código del juego. Este tipo está dirigido esencialmente a los profesionales independientes: Unity 3D, Torque 3D.

Los motores profesionales están dirigidos a los profesionales con gran presupuesto. Son los motores más completos y los más potentes del mercado, se mantienen al día por sus creadores y disponen de los últimos avances tecnológicos: Id Tech 4, Unreal Engine 3, Cry Engine 3.

Unity 3D

Unity 3D es un motor de creación de videojuegos 3D lanzado oficialmente como tal el 1 de Junio 2005. Este motor permite la creación de juegos y otros contenidos interactivos como diseños arquitectónicos o animaciones 3D en tiempo real.

Muchas personas interesadas por el desarrollo se topan con la dificultad de aprender los lenguajes de programación y los motores que los utilizan. Sin estudios de programación o de animación por ordenador, el aprendizaje de los conceptos, métodos y los principios necesarios para la creación de un videojuego se hace muy difícil.

Unity pone la potencia de su motor al servicio de los utilizadores permitiéndoles obtener un resultado de máxima calidad con un mínimo de esfuerzo.

Unity existe en versión profesional que se puede adquirir previo pago y una versión libre completamente gratuita que se puede descargar en la página Web de Unity. Esta última versión incluye menos funcionalidades pero aun así permite la creación de programas con muy buena calidad.

Unity es una aplicación 3D en tiempo real y multimedia además de ser motor 3D y físico utilizado para la creación de juegos en red, de animación en tiempo real, de contenido interactivo compuesto por audio, video y objetos 3D. Este motor no permite la modelización pero permite crear escenas que soportan iluminación, terrenos, cámaras, texturas[Ouazzani, 2012].

Unity presenta varias ventajas que hacen que sea uno de los motores de videojuego más cotizado del momento. En los siguientes párrafos se van a ir citando todas estas ventajas:

* Permite la importación de numerosos formatos 3D como 3ds Max, Maya, Cinema 4D, Cheetah3D y Softimage, Blender, Modo, ZBrush, FBX o recursos variados tales como texturas Photoshop, PNG, TIFF, audios y videos.
* Es compatible con las API gráficas de Direct3D, OpenGL y Wii. Además de ser compatible con QuickTime y utilizar internamente el formato Ogg Vorbis.
* Este software tiene la particularidad de incluir la herramienta de desarrollo MonoDevelop con la que se pueden crear scripts en JavaScript, C# y un dialecto de Python llamado Boo con los que extender la funcionalidad del editor, utilizando las API que proveé y la cual encontramos documentada junto a tutoriales y recursos en su web oficial.
* La estructura creada por Unity viene definida mediante escenas que representan alguna parte del juego.
* La aplicación permite acceder al Asset Store donde existe multitud de recursos gratuitos y de pago. Incluso se puede extender la herramienta mediante plugins que se obtienen en esta misma tienda.
* Dispone de una interfaz de desarrollo muy bien definida e intuitiva que permite.
* Existe en varias versiones en función de los módulos elegido, la versión más simple destinada a los amateurs es gratuita.

Unity resulta ser un herramienta que permite a sus desarrolladores obtener productos de manera sencilla y rápida.

Técnicas utilizadas en la obtención de requerimientos

Existen varias técnicas para la obtención de requerimiento, sin embargo, se mencionarán algunas de las más importantes. La técnica más apropiada para cada actividad dependerá del proyecto que esté desarrollándose.

Entrevistas y Cuestionarios

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado; el cuestionario consiste en una serie de preguntas relacionadas con varios aspectos de un sistema.

Por lo común, los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él.

Las preguntas que deben realizarse en esta técnica, deben ser preguntas de alto nivel y abstractas que pueden realizarse al inicio del proyecto para obtener información sobre aspectos globales del problema del usuario y soluciones potenciales.

Con frecuencia, se utilizan preguntas abiertas para descubrir sentimientos, opiniones y experiencias generales, o para explorar un proceso o problema. Este tipo de preguntas son siempre apropiadas, además que ayudan a entender la perspectiva del afectado y no están influenciadas por el conocimiento de la solución.

El éxito de esta técnica combinada, depende de la habilidad del entrevistador y de su preparación para la misma. Los analistas necesitan ser sensibles las dificultades que algunos entrevistados crean durante la entrevista y saber cómo tratar con problemas potenciales. Asimismo, necesitan considerar no sólo la información que adquieren a través del cuestionario y la entrevista, sino también, su significancia [Pérez Huebe, 2005].

Lluvia de ideas

Este método comenzó en el ámbito de las empresas, aplicándose a temas tan variados como la productividad, la necesidad de encontrar nuevas ideas y soluciones para los productos del mercado, encontrar nuevos métodos que desarrollen el pensamiento creativo a todos los niveles, etc.

Pero pronto se extendió a otros ámbitos, incluyendo el mundo de desarrollo de sistemas; básicamente se busca que los involucrados en un proyecto desarrollen su creatividad, promoviendo la introducción de los principios creativos [Pérez Huebe, 2005].

Principios de la lluvia de ideas

* Aplazar el juicio y no realizar críticas, hasta que no se agoten las ideas, ya que actuaría como un inhibidor. Se debe crear una atmósfera de trabajo en la que nadie se sienta amenazado.
* Cuantas más ideas se sugieren, mejores resultados se conseguirán: “la cantidad produce la calidad”. Las mejores ideas aparecen tarde en el periodo de producción de ideas, será más fácil que se encuentren las soluciones y sé tendrá más variedad sobre la que hay que elegir.
* La producción de ideas en grupos puede ser más efectiva que la individual.
* Tampoco se debe olvidar que durante las sesiones, las ideas de una persona, serán asociadas de manera distinta por cada miembro, y hará que aparezcan otras por contacto.

Las personas que componen el grupo deben estar motivadas para solucionar el problema, y con un ambiente que propicie la participación de todos. Pueden sentirse confiados y con la sensación de que pueden hablar sin que se produzcan críticas. Todas las ideas en principio deben tener el mismo valor, pues cualquiera de ellas puede ser la clave para la solución. Es necesario prestar mucha atención a las frases que pueden coartar la producción de ideas.

Además durante la celebración no deben asistir espectadores. Se deben evitar todos los bloqueos que paralizan la idea: como son los hábitos o ideas preconcebidas, el desánimo o falta de confianza en sí mismo, el temor y la timidez [Pérez Huebe, 2005].

Prototipos

Los prototipos permiten al desarrollador crear un modelo del software que debe ser construido.

Al igual que todos los enfoques al proceso de desarrollo del software, el prototipado comienza con la captura de requerimientos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos, y señalan áreas en las que será necesaria la profundización en las definiciones.

Luego de esto, tiene lugar un "diseño rápido". El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario (por ejemplo, entradas y formatos de las salidas). El diseño rápido lleva la construcción de un prototipo.

El prototipo es evaluado por el cliente y el usuario, y utilizado para refinar los requerimientos del software a ser desarrollado. Un proceso de iteración tiene lugar a medida que el prototipo es “puesto a punto" para satisfacer las necesidades del cliente y permitiendo al mismo tiempo una mejor comprensión del problema por parte del desarrollador [Pérez Huebe, 2005].

Existen principalmente dos tipos de prototipos:

Prototipo rápido: El prototipado rápido es un mecanismo para lograr la validación pre-compromiso. Se utiliza para validar requerimientos en una etapa previa al diseño específico. En este sentido, el prototipo puede ser visto como una aceptación tácita de que los requerimientos no son totalmente conocidos o entendidos antes del diseño y la implementación. El prototipo rápido puede ser usado como un medio para explorar nuevos requerimientos y así ayudar a "controlar" su constante evolución.

.

Prototipo evolutivo: Desde una perspectiva diferente, todo el ciclo de vida de un producto puede ser visto como una serie incremental de detallados prototipos acumulativos. Tradicionalmente, el ciclo de vida está dividido en dos fases distintas: desarrollo y mantenimiento.

La experiencia ha demostrado que esta distinción es arbitraria y va en contra de la realidad ya que la mayor parte del costo del software ocurre después de que el producto se ha entregado.

El punto de vista evolutivo del ciclo de vida del software considera a la primera entrega como un prototipo inicial en el campo. Modificaciones y mejoras subsecuentes resultan en nuevas entregas de prototipos más maduros.

Este proceso continúa hasta que se haya desarrollado el producto final. La adopción de esta óptica elimina la distinción arbitraria entre desarrollo y mantenimiento, resultando en un importante cambio de mentalidad que afecta las estrategias para la estimación de costos, enfoques de desarrollo y adquisición de productos [Pérez Huebe, 2005].

Casos de Uso

Los casos de uso son una técnica para la especificación de requerimientos funcionales. Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra.

Los casos de uso son una técnica para especificar el comportamiento de un sistema: “Un caso de uso es una secuencia de interacciones entre un sistema y actores que usan alguno de sus servicios”.

Los actores son personas u otros sistemas que interactúan con el sistema cuyos requerimientos se están describiendo. Un actor puede participar en varios casos de uso y un caso de uso puede estar relacionado con varios actores.

Los casos de uso presentan ciertas ventajas sobre la descripción meramente textual de los requerimientos funcionales, ya que facilitan la elicitación de requerimientos y son fácilmente comprensibles por los clientes y usuarios. Además, pueden servir de base a las pruebas del sistema y a la documentación para los usuarios.