



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**
título del TFG



Presentado por Francisco Martín Vargas
en Universidad de Burgos — 4 de septiembre
de 2021

Tutor: D. César Represa Pérez



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. César Represa Pérez, profesor del departamento de Ingeniería Electromecánica, área de Tecnología Electrónica.

Expone:

Que el alumno D. Francisco Martín Vargas, con DNI 71704736M, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 4 de septiembre de 2021

Vº. Bº. del Tutor:

D. César Represa Pérez

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	4
2.3. Objetivos personales	4
Conceptos teóricos	5
3.1. Android	5
3.2. Python	6
3.3. Domótica	7
3.4. IoT	7
3.5. Dispositivos de bajo consumo	7
Técnicas y herramientas	9
4.1. Herramientas de control de versiones	9
4.2. Herramientas de gestión de proyectos	10
4.3. Metodologías	10
4.4. Herramientas de evaluación de código	10
4.5. Herramientas de documentación	10
4.6. Lenguajes de programación	11

4.7. Entornos de desarrollo integrado (IDE)	12
4.8. Herramientas de automatización de compilación del código	12
4.9. Protocolos de comunicación	13
4.10. Librerías	14
Aspectos relevantes del desarrollo del proyecto	17
5.1. Elección del tema	17
5.2. Comienzo del proyecto	17
5.3. Desarrollo de la aplicación y servidor	18
Trabajos relacionados	19
Conclusiones y Líneas de trabajo futuras	21
Bibliografía	23

Índice de figuras

3.1. Sistemas operativos mas usados desde 2009 a 2021 [5]	6
3.2. lenguajes de programación mas usados desde 2012 a 2018 [3] . .	7

Índice de tablas

Introducción

Hoy en día en los hogares cada vez estamos más rodeados de tecnologías, tanto móviles, como aparatos del hogar, los cuales disponen de sistemas informáticos muy avanzados y automatizados que nos proporcionan herramientas para ser manejados a distancia. Hablamos de lavadoras, frigoríficos, enchufes, interruptores, termostatos, e incluso aspiradores inteligentes, todos ellos monitorizables a distancia.

La domótica es el conjunto de todas las tecnologías que agrupan el control y la automatización inteligente de una vivienda. Gracias a ella llegaremos a gestionar de una manera más eficiente nuestros recursos, además de aportarnos más seguridad y un extra de confort.

El sector domótico está creciendo a gran velocidad hoy en día y ha evolucionado notablemente recientemente. Gracias a esta evolución y crecimiento, podemos obtener más funcionalidad, facilidad de uso e instalación, variedad de producto, además de mayor calidad y mayor oferta.

Pero no sólo ha ayudado a las personas, gracias a la eficiencia energética y de recursos, también colabora con el cambio climático. Gracias a este sector, se reducen los consumos en calefacción, en energía para la vivienda, en gasto de cantidad de agua, etc, todo ello contribuye a detener el cambio climático además de a la población. En resumen, la domótica ha ayudado a mejorar la calidad de vida de las personas.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

El objetivo principal que persigue este proyecto es el desarrollo de un entorno domótico que permita la visualización de cámaras en el hogar a través de una aplicación Android, todo ello a través de la comunicación entre un servidor que realice la gestión y el propio dispositivo Android.

El servidor debe establecer la conexión con las cámaras y ser capaz de retransmitir la imagen al dispositivo Android, conectado a la misma red WiFi.

2.1. Objetivos generales

- Desarrollar una aplicación para el entorno Android que permita la monitorización de cámaras dispuestas en un hogar.
- Desarrollar y desplegar un servidor que gestione el entorno domótico (la conexión con las cámaras y con la aplicación Android).
- Facilitar al usuario la monitorización mediante una interfaz fácil y sencilla.
- Reutilizar dispositivos en desuso.
- Utilizar dispositivos de bajo consumo.

2.2. Objetivos técnicos

- Desarrollar un servidor con el lenguaje Python que gestione toda la complejidad.
- Desarrollar una aplicación en el entorno Android con soporte para API 16 y superiores.
- Emplear la herramienta Gradle para la automatización del proceso de construcción y compilación de software de la aplicación Android.
- Utilizar Git como sistema de control de versiones distribuido junto con la plataforma GitHub.
- Aprovechar las posibilidades que ofrecen las herramientas de integración continua como SonarQube en el repositorio.

2.3. Objetivos personales

- Mostrar la capacidad de los conocimientos adquiridos durante el periodo universitario.
- Aprender a manejar metodologías y herramientas innovadoras que son utilizadas en el mercado laboral.
- Mejorar mi habilidad en el desarrollo de aplicaciones en el entorno Android.
- Profundizar en los conceptos aprendidos sobre Python en mi formación universitaria.

Conceptos teóricos

Para comprender el marco teórico del desarrollo de este proyecto, debemos conocer previamente algunos conceptos en los que se basa este propósito.

Seguir con una pequeña introducción.

3.1. Android

Android [4] es un sistema operativo (OS) que inicialmente fue desarrollado para dispositivos móviles pero que hoy en día engloba ordenadores, relojes, televisores, tablets, coches, etc. Este entorno ha sido desarrollado por Google cuyo objetivo fue fomentar el uso de una plataforma abierta, gratuita, multiplataforma y muy segura, y es por ello que está basado en Linux (Linux es un núcleo sistema operativo (OS) open source).

Este sistema permite programar aplicaciones empleando una variación de Java llamada Dalvik (o ART a partir de su versión 5.0) y proporciona todas las interfaces necesarias para desarrollar fácilmente aplicaciones que acceden a las funciones del dispositivo (como pudiera ser el GPS, o la memoria entre otros).

Y es por todo esto que es el sistema operativo más usado el mundo.

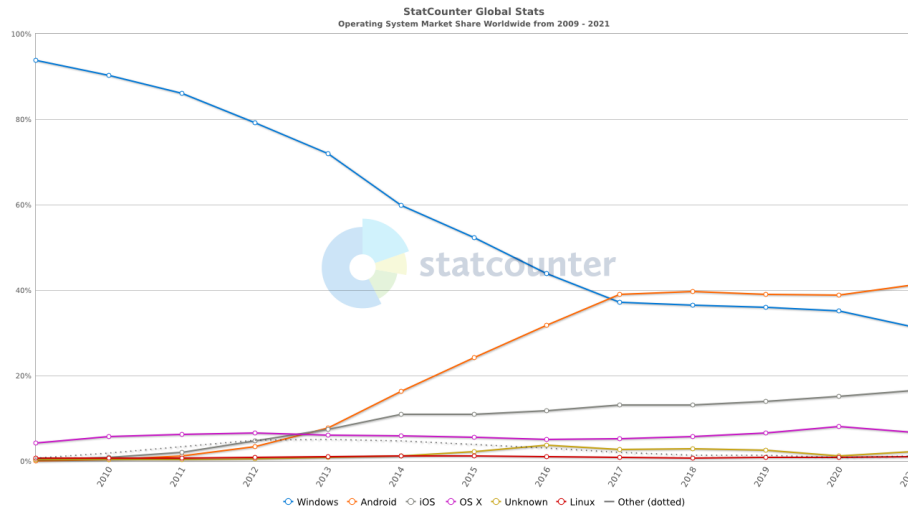


Figura 3.1: Sistemas operativos mas usados desde 2009 a 2021 [5]

3.2. Python

Python [2] es un lenguaje interpretado, interactivo y orientado a objetos, que incorpora módulos, excepciones, tipado dinámico, tipos de datos de muy alto nivel y clases. Está muy extendido y nos permite realizar cualquiera de nuestros propósitos gracias a su gran cantidad de librerías y programadores. Fue creado a principios de la década de 1990 por Guido van Rossum en Stichting Mathematisch Centrum (CWI) en los Países Bajos como sucesor de un idioma llamado ABC [1].

Este lenguaje [2] nos proporciona una gran biblioteca estándar que abarca áreas como procesamiento de cadenas (ya sea tanto expresiones regulares, Unicode, o incluso cálculo de diferencias entre archivos), protocolos de Internet (como son HTTP, FTP, SMTP, XML-RPC, POP, IMAP, o programación CGI entre otros), ingeniería de software (desde pruebas unitarias, registro, creación de perfiles, hasta análisis del propio código Python) e interfaces del sistema operativo (tales como llamadas al sistema, sistemas de archivos, incluidos sockets TCP / IP).

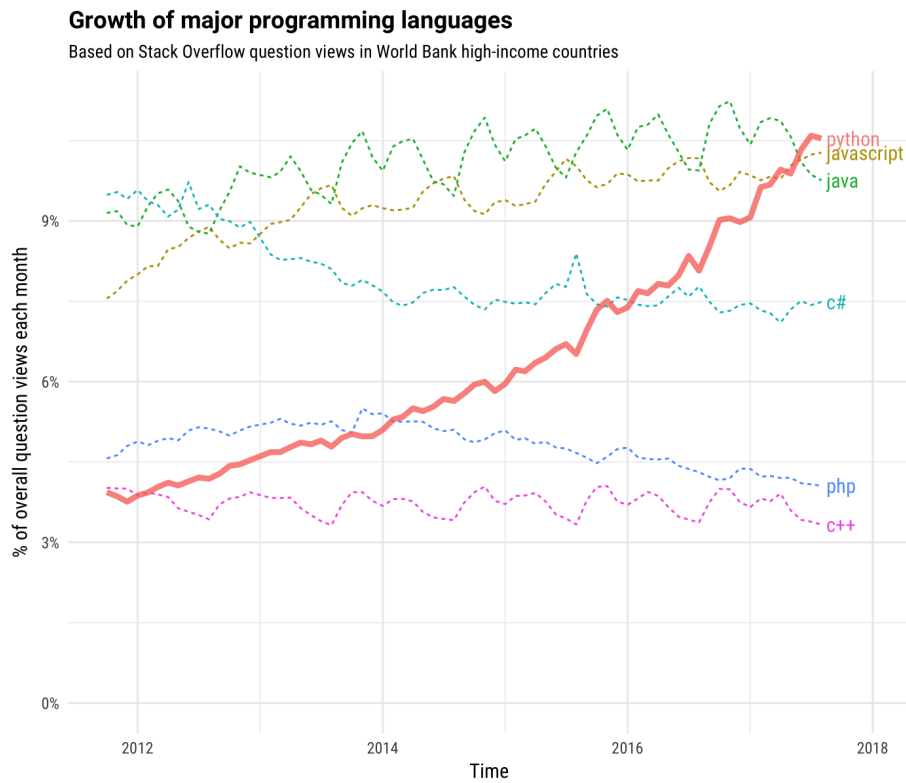


Figura 3.2: lenguajes de programación mas usados desde 2012 a 2018 [3]

3.3. Domótica

Domótica

3.4. IoT

IoT

3.5. Dispositivos de bajo consumo

Raspberry Pi

Técnicas y herramientas

4.1. Herramientas de control de versiones

Github

GitHub es una plataforma que nos permite gestionar y organizar nuestros proyectos y está basada en la nube, incorporando las funciones de control de versiones que proporciona Git. Esta herramienta posibilita a los desarrolladores poder almacenar y administrar su código, realizar un registro y control de los cambios sobre el código almacenado. Es una de las herramientas más populares entre los desarrolladores, cuenta con más de 100 millones de repositorios, y la mayoría de ellos son de código abierto.

He utilizado GitHub para el alojamiento de mi proyecto en el repositorio “<https://github.com/fmv1001/LocalStream>”.

Git Bash

Git Bash es una aplicación para Windows para la emulación de la línea de comandos de Git a través de una shell. Una shell es un intérprete de comandos que provee una interfaz de usuario para la comunicación con el sistema operativo.

Esta aplicación me ha servido para subir cambios al repositorio de GitHub a través de los comandos git.

4.2. Herramientas de gestión de proyectos

ZenHub

ZenHub es una plataforma para la gestión ágil de proyectos que se integra con github, funcionando como aplicación nativa en su interfaz. Te ayuda a planificar tu proyecto dentro de GitHub, automatiza el flujo de trabajo. Más del 75 % de los desarrolladores que usan ZenHub en sus proyectos dicen que ZenHub mejora su enfoque y ayuda en el envío de un mejor software en un menor periodo de tiempo, y el 65 % informan de proyectos con un mejor alcance.

ZenHub me ha permitido gestionar mi proyecto, ayudándome con la planificación del mismo, para cumplir los tiempos de entrega del mismo.

4.3. Metodologías

Modelo Cliente-Servidor

Modelo Cliente-Servidor.

4.4. Herramientas de evaluación de código

SonarQube

SonarQube es una plataforma de código abierto desarrollada en Java que nos permite realizar análisis de código con diferentes herramientas de forma automatizada. Usa diferentes herramientas de análisis estático de código fuente como pueden ser Checkstyle, PMD o FindBugs para obtener métricas que pueden ayudar a mejorar la calidad del código de un programa.

En este proyecto se ha usado la herramienta Sonar-Scanner de SonarQube, para la evaluación del código tanto de la aplicación Android en java como del servidor desarrollado en python.

4.5. Herramientas de documentación

L^AT_EX

L^AT_EX es un sistema de software libre de composición tipográfica de alta calidad, orientado a la producción de documentación técnica y científica.

L^AT_EX es el estándar de facto para la comunicación y publicación de documentos científicos gracias a sus características, posibilidades y calidad profesional.

He usado L^AT_EX para el desarrollo de tanto este documento como de los anexos.

4.6. Lenguajes de programación

Python

Python es un lenguaje de programación ampliamente utilizado por empresas de casi todo el mundo para la construcción de aplicaciones web, análisis de datos, automatización de operaciones y creación de aplicaciones empresariales con alta fiabilidad y escalabilidad.

Para desarrollar el servidor he utilizado la versión 3.0 del lenguaje Python.

Java

Java es un lenguaje de programación orientado a objetos y basado en clases que se usa para el desarrollo de aplicaciones, diseñado para tener la menor cantidad de dependencias de implementación posibles. Además es uno de los lenguajes de programación utilizados para el desarrollo de aplicaciones para el sistema operativo Android. El código compilado de Java puede ejecutarse en cualquier plataforma que tenga instalada la máquina virtual java, sin la necesidad de volver a ser compilado.

Java es el lenguaje que he escogido para la construcción de la aplicación Android debido a mi familiaridad con este lenguaje.

SQL

SQL (Structured Query Language) es un lenguaje declarativo que ha sido diseñado para la administración y recuperación de información de sistemas de gestión de bases de datos relacionales. Este lenguaje es conservado por el organismo ANSI (American National Standards Institute).

El lenguaje SQL ha sido muy útil en este proyecto a la hora de respaldar al información necesaria tanto en el servidor como en la aplicación.

4.7. Entornos de desarrollo integrado (IDE)

Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para el sistema operativo Android y está basado en IntelliJ IDEA (IDE de Java desarrollado por JetBrains). Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, entre otras tenemos: un sistema de compilación flexible basado en Gradle o la integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.

Android Studio es el IDE oficial para desarrollar aplicaciones en Android y por eso lo he escogido para mi proyecto.

Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Es gratuito y de código abierto aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Para el despliegue del servidor he utilizado esta aplicación open source de Microsoft.

4.8. Herramientas de automatización de compilación del código

Gradle

Gradle, es una herramienta que permite la automatización de compilación de código abierto, la cual se encuentra centrada en la flexibilidad y el rendimiento. Los scripts de compilación de Gradle se escriben utilizando Groovy o Kotlin DSL (Domain Specific Language). Gradle tiene una gran flexibilidad y nos deja hacer usos otros lenguajes y no solo de Java, también cuenta con un sistema de gestión de dependencias muy estable. Además es el sistema de compilación oficial para Android y cuenta con soporte para diversas tecnologías y lenguajes.

4.9. Protocolos de comunicación

UDP

El protocolo de datagramas de usuario (User Datagram Protocol o UDP) es un protocolo de la capa de transporte basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

Gracias a este protocolo el servidor envía las imágenes al dispositivo móvil Android sin que este tenga que confirmar que lo ha recibido.

TCP / IP

TCP es un protocolo de capa de transporte utilizado por aplicaciones que requieren una entrega garantizada. Se trata de un protocolo de ventana deslizante que permite gestionar tanto los tiempos de espera como las retransmisiones. TCP establece una conexión virtual full duplex entre dos puntos finales. Cada punto final está definido por una dirección IP y un número de puerto TCP. El funcionamiento de TCP se implementa como una máquina de estado finito.

Para que el servidor y la aplicación se comuniquen a la hora de realizar alguna tarea, este protocolo ha permitido que en ningún momento se pierda la conexión y en ese caso los dos serán conscientes y actuarán para salvaguardar la información y mantener un sistema coherente.

Sockets

Los sockets hacen posible la comunicación entre conectores de procesos, cada uno de ellos tiene asociada una dirección, un puerto, y un protocolo asociado (UDP o TCP). Los sockets nos permiten implementar una arquitectura cliente-servidor. La comunicación debe ser iniciada por uno de los procesos que se denomina programa cliente". El segundo proceso espera a que otro inicie la comunicación, por este motivo se denomina programa "servidor". Un socket es un proceso o hilo existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

He usado los sockets junto a los protocolos mencionados en esta misma sección para la comunicación entre el servidor y el cliente (dispositivo Android).

4.10. Librerías

Sqlalchemy

SQLAlchemy es el conjunto de herramientas SQL de Python y el mapeador relacional de objetos que ofrece a los desarrolladores de aplicaciones toda la potencia y flexibilidad de SQL. Proporciona un conjunto completo de patrones de persistencia de nivel empresarial bien conocidos, diseñados para un acceso a la base de datos eficiente y de alto rendimiento, adaptados a un lenguaje de dominio sencillo y pitónico.

OpenCV

OpenCV es una enorme biblioteca de código abierto para la visión por ordenador, el aprendizaje automático y el procesamiento de imágenes. OpenCV es compatible con una gran variedad de lenguajes de programación como Python, C++, Java, entre otros.

OpenCV me ha permitido mantener una conexión con las cámaras disponibles y a su vez obtener las imágenes de estas.

SQLite

SQLite es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta fiabilidad y con todas las funciones. SQLite es el motor de base de datos más utilizado en el mundo. SQLite está integrado en todos los teléfonos móviles y en la mayoría de los ordenadores, y viene incluido en innumerables aplicaciones que la gente utiliza a diario. SQLite es la API para utilizar una base de datos en Android.

Android Support Library

El paquete Android Support Library es un conjunto de bibliotecas de código que proporcionan versiones compatibles con versiones anteriores de las API del marco de trabajo de Android, así como características que sólo están disponibles a través de las API de la biblioteca. Cada biblioteca de apoyo es compatible con un nivel específico de la API de Android.

AndroidX

AndroidX es el proyecto de código abierto que el equipo de Android utiliza para desarrollar, probar, empaquetar, versionar y liberar bibliotecas dentro de Jetpack. AndroidX es una importante mejora de la biblioteca de soporte original de Android.

JavaFx

JavaFX es un conjunto de paquetes gráficos y multimedia que permite a los desarrolladores diseñar, crear, probar, depurar e implantar aplicaciones cliente enriquecidas que funcionan de forma coherente en diversas plataformas. El SDK de JavaFX-Android contiene una implementación de JavaFX que se ejecuta en Android, junto con algunas herramientas para construir los paquetes de Android.

JUnit

Unit es un marco de pruebas unitarias para el lenguaje de programación Java. Desempeña un papel crucial en el desarrollo dirigido por pruebas, y es una familia de marcos de pruebas unitarias conocidos colectivamente como xUnit. JUnit es un framework de "Unit Testing" para aplicaciones Java que ya está incluido por defecto en android studio.

Material Design

Material Design es un lenguaje de diseño orientado a Android creado por Google, que admite experiencias táctiles en pantalla a través de funciones ricas en pistas y movimientos naturales que imitan los objetos del mundo real. Material design es una guía completa para el diseño visual, de movimiento y de interacción en todas las plataformas y dispositivos.

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación.

5.1. Elección del tema

Siempre me ha gustado el hecho de desarrollar una aplicación Android y cuando estuve mirando tema para mi proyecto, me llamó la atención el hecho de desarrollar algo que me sirviera en mi día a día, se ofertaba un trabajo para hacer una aplicación con Unity3D, pero el profesor había cambiado de área y ya no lo tutorizaba. Así que decidí darle mi enfoque y realizar la aplicación para el sistema Android dado lo extendido que está. También decidí hacer el servidor dado que una asignatura que he cursado en este último curso trataba el tema y me llamó mucho la atención. Y todo esto me llevó a realizar el proyecto que expongo a continuación.

5.2. Comienzo del proyecto

Una vez escogido el tema, tocaba diseñar la estructura del proyecto. Decidí que la mejor opción sería separar el servidor en un dispositivo y la aplicación en otro, ya que así el dispositivo no cargaría con toda la carga computacional que supondría para él, dado que estos no disponen de mucha batería, aplicando el modelo cliente-servidor. Ahora tocaba decidir donde desarrollar el servidor y en que lenguaje. En

este paso, me decanté por el lenguaje Python ya que está muy extendido (llegando a ser uno de los lenguajes de programación mas usado del mundo [3]) y estaba muy familiarizado con él debido a que lo he usado en numerosas ocasiones.

Después tenía que asegurarme de poder realizar la conexión con una cámara a través de Python. Tras una búsqueda intensa por internet encontré la librería OpenCV. Para la realización de este proyecto me compré una cámara IP, pero tras unos días no conseguía acceder a ella desde OpenCV con python. Para no perder demasiado tiempo contacté con la empresa que me había vendido la cámara y les comenté mi problema. Rápidamente me contestaron y me proporcionaron la dirección que debía usar, y gracias a ello pude realizar una conexión servidor-cámara.

Una vez realizada dicha conexión ya podía empezar con el siguiente punto del proyecto.

5.3. Desarrollo de la aplicación y servidor

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Python Software Foundation. Historia y licencia, 2021. [Internet; accedido 01-septiembre-2021].
- [2] Python Software Foundation. Preguntas frecuentes generales sobre python, 2021. [Internet; accedido 01-septiembre-2021].
- [3] David Robinson. The incredible growth of python, 2017. [Internet; accedido 01-septiembre-2021].
- [4] David Robledo. *Desarrollo de aplicaciones para Android I*. Ministerio de Educación, Cultura y Deporte, 2016.
- [5] Statcounter. Operating system market share worldwide - august 2021, 2021. [Internet; accedido 01-septiembre-2021].