



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

**Aplicación de soporte a personas con
dificultades de visión**



Presentado por Bryan Reinoso Cevallos
en Universidad de Burgos — Julio de 2014
Tutores: José Francisco Díez Pastor,
Dr. César I. García Osorio

Resumen

Descriptores

Abstract

Keywords

Índice general

1. Introducción	1
2. Objetivos del proyecto	3
2.1. Objetivos funcionales	3
2.1.1. Objetivos funcionales en el cliente Android	3
2.1.2. Objetivos funcionales en el servidor Flask	3
3. Conceptos teóricos	5
4. Técnicas y herramientas	7
4.1. Técnicas de desarrollo	7
4.1.1. Metodología de desarrollo ágil	7
4.2. Herramientas utilizadas	7
4.2.1. Gestor de Tareas: VersionOne	7
4.2.2. Gestor de Versiones: Git Hub	7
4.2.3. IDEL de desarrollo: Android Studio	8
4.2.4. Herramienta de Generación de Documentación: \LaTeX	8
4.2.5. Herramientas de Deep Learning: NeuralTalk	8
4.2.6. Herramientas de Deep Learning: Caffe	9
4.2.7. Herramientas de programación: MATLAB	9
4.2.8. Herramientas de desarrollo de servidores: Flask	9
4.2.9. Manual del Programador	9
5. Estado del arte	15
6. Aspectos relevantes del desarrollo del proyecto	17
6.1. Resolviendo puzles en un ordenador	17
7. Conclusiones y líneas de trabajo futuras	19
7.1. Dificultades encontradas	19
7.1.1. Dificultades con GSOAP y Apache	19
7.1.2. Dificultades con DeepBeliefSDK	20
7.1.3. Dificultades en la instalación de Herramientas	20
7.1.4. Dificultades con NeuralTalk	20
7.1.5. Dificultades con Android	20
7.2. Conclusiones del Proyecto	21
7.3. Líneas de trabajo Futuras	21
Anexos	21

Índice de figuras

4.1. Página de descarga del JDK de Java	10
4.2. Instalación del JDK, paso 1.	11
4.3. Instalación del JDK, paso 2.	12
4.4. Instalación del JDK, paso 3.	13
4.5. Instalación del JDK, paso 4.	14

Índice de tablas

1. INTRODUCCIÓN

El objetivo principal del proyecto consiste en la creación de una aplicación de soporte para personas con dificultades de visión.

La aplicación de soporte para los usuarios será desarrollada en Android y constará de una interfaz y funcionamiento orientada al uso por personas con dificultades de visión, además de guiará al usuario a través de la aplicación con un asistente de voz que irá diciendo al usuario qué debe hacer y en qué punto de la aplicación se encuentra.

La aplicación tendrá como objetivo inicial mandar una foto tomada por el usuario a un servidor que la procesará y extraerá una predicción que le dirá al usuario lo que se puede observar en la imagen que él ha elegido. La predicción será una frase que describa la imagen y esta será leída por la aplicación.

Por otra parte en el lado del servidor usaremos algoritmos de machine learning para el procesamiento de la imagen y la extracción de la predicción. El servidor recibirá una petición de tipo POST y cargará la imagen, la cual será procesada y en respuesta devolverá una frase en español que describa la imagen.

El principal objetivo del proyecto es conseguir crear un prototipo de esta aplicación que tenga un correcto funcionamiento y que sirva como base para desarrollar sobre él una aplicación más optimizada y de aún mejor funcionamiento.

2. OBJETIVOS DEL PROYECTO

2.1 Objetivos funcionales

El principal objetivo del proyecto es tener un prototipo de la aplicación que funcione de manera adecuada, para esto tenemos que tener en cuenta tanto los objetivos del lado del servidor como del lado del cliente.

2.1.1 Objetivos funcionales en el cliente Android

El principal objetivo que podemos encontrarnos en el lado del cliente es el disponer de un prototipo de aplicación Android con la que el usuario pueda mandar de manera intuitiva una imagen al servidor y recibir de él la predicción esperada. Para que esto funcione de manera adecuada se han propuesto una serie de requisitos que deben cumplirse:

- Para poder enviar la imagen al servidor y recibir de él la predicción, se debe establecer una conexión con el desde la aplicación Android y hacer una correcta petición de tipo POST. Para ello usamos la librería de Apache con el módulo MultipartEntityBuilder, que nos permite establecer de manera bastante sencilla la conexión y mandar la petición POST. Además por especificaciones de Android esto deberá realizarse en un hilo separado del hilo principal.
- Para que la persona que lo utilice no tenga por qué saber la estructura de la interfaz gráfica de la aplicación para poder usarla sin problemas, nos apoyaremos sobre los eventos ontouch de Android y evitaremos lo máximo posible que el usuario tenga que saber cómo es la interfaz.
- Para ofrecer una guía fácil y útil para el usuario a lo largo de su experiencia usando la aplicación, usaremos la librería Text2Speech para ir guiando al usuario con instrucciones en voz alta explicándole lo que debe hacer en cada momento. Finalmente le devolveremos la predicción y se le dirá cuáles son sus opciones.
- Para poder tomar la imagen se requerirá de un dispositivo con cámara de fotos y se tendrá que establecer los permisos sobre la aplicación para el uso de la misma.

2.1.2 Objetivos funcionales en el servidor Flask

En el lado del servidor tenemos un objetivo claro y es el de recibir una imagen a través de una petición POST desde un cliente, procesarla y, finalmente, devolver la predicción en español. Para conseguir todo esto tenemos una serie de requisitos y objetivos que tenemos que cumplir:

- Para poder recibir las peticiones y mandar respuestas tenemos que tener un servidor con la capacidad de gestionar este tipo de operaciones, para ello se ha decidido usar Flask para programar el servidor. Con Flask se creará un servidor que gestione la petición post sobre un URL concreto .

- Para poder procesar la imagen se necesita una red neuronal a través de la cual pasar la imagen y extraer la predicción. Para ello se ha elegido un proyecto desarrollado por **Karpathy** y su proyecto **NeuralTalk**. Para ello se instalará el proyecto en la máquina servidora y todas las dependencias del mismo. Además se hará todas las modificaciones necesarias para su correcto funcionamiento sobre nuestra aplicación.
- Una vez la predicción haya sido extraída esta viene en inglés, así que el servidor se deberá encargar de traducirla al español para enviarla al cliente. Para ello se usa una conexión con el traductor de Google.

3. CONCEPTOS TEÓRICOS

En este apartado se profundizará...

4. TÉCNICAS Y HERRAMIENTAS

En este apartado se indicarán las técnicas y herramientas utilizadas durante la realización del proyecto.

4.1 Técnicas de desarrollo

En esta sección se indicarán las técnicas de desarrollo utilizadas a lo largo del proyecto.

4.1.1 Metodología de desarrollo ágil

4.2 Herramientas utilizadas

En este apartado se mostrará las distintas herramientas utilizadas para el desarrollo del proyecto.

4.2.1 Gestor de Tareas: VersionOne

Se ha estudiado entre varias posibles herramientas, entre ellas están:

- PivotalTracker
- FogBugz
- VersionOne

Se ha optado por la herramienta VersionOne, que ofrece unas condiciones notablemente mejores a las otras en su versión gratuita y además resulta bastante intuitiva y fácil de usar.

Con esta herramienta nos encargaremos de generar los Sprints del proyecto y se gestionará las tareas dentro de cada uno. Esta herramienta es usada en el desarrollo ágil, que es el tipo de desarrollo que se llevará a cabo en el proyecto, además de que con la herramienta podemos, posteriormente, generar una serie de gráficos e informes que nos serán de gran ayuda a la hora de documentar el proyecto y de gestionar el avance del mismo.

4.2.2 Gestor de Versiones: Git Hub

Se ha estudiado entre varias posibles herramientas, entre ellas están:

- GitHub
- Bitbucket

- Sourcefroge

Finalmente se decidió que se iba a usar la herramienta GitHub porque se tenía experiencia previa en el uso de la misma, ofrece unas condiciones bastante razonables en su versión gratuita y se puede hacer un buen seguimiento del proyecto con ella. Además de que es bastante sencilla la generación de commits en Windows porque dispone de una aplicación con la que los commits se realizan de manera sencilla. Se puede ver de una manera bastante clara cómo ha ido avanzando el proyecto y se puede extraer unos gráficos muy útiles a la hora de documentar y determinar cómo ha avanzado el proyecto. Además al estar en su versión gratuita los miembros del jurado y cualquier persona que desee el acceso al proyecto sólo necesitará el link del mismo para poder verlo y comprobar la asiduidad de los commits y cómo ha sido la evolución del proyecto.

4.2.3 IDEL de desarrollo: Android Studio

Se ha estudiado entre varias posibles herramientas, entre ellas están:

- Eclipse
- Android Studio

La elección de Android Studio ha sido porque no sólo es una herramienta exclusivamente dedicada a aplicaciones Android, sino que resultaba más prometedora que Eclipse; la cuál pensamos que puede quedar obsoleta para este tipo de aplicaciones. También vemos que Android Studio ofrece un gestor de instalación de paquetes de kit de desarrollo y plugins, lo cuál puede resultar muy útil a la hora de la configuración del entorno de desarrollo. Además en Android Studio no puede sólo usarse máquinas virtuales de móviles, sino que puedes conectar un dispositivo móvil al ordenador e ir ejecutando tu aplicación sobre el contando con un debugger y un logcat para errores, lo cual facilita en gran medida la programación.

4.2.4 Herramienta de Generación de Documentación: \LaTeX

Se ha elegido esta herramienta debido a su facilidad de uso y a que optimiza automáticamente la estructura del producto final para ofrecer el mejor resultado visual. También prevemos que el uso de esta herramienta facilitará en gran medida la carga de trabajo al realizar la documentación lo que permitirá ahorrarse tiempo en este apartado, producir un resultado mas eficiente y bueno y, finalmente, se podrá usar el tiempo ahorrado en otras cosas para el avance del proyecto.

4.2.5 Herramientas de Deep Learning: NeuralTalk

Se ha estudiado entre varias posibles herramientas, entre ellas están:

- Lib CCV
- Overfeat
- Deep Belief SDK

Esta herramienta ha sido elegida porque venía en el mismo idioma en el que se iba a programar el servidor, además de que su aplicación en nuestro proyecto era mucho mas práctico, pues devuelve una frase descriptiva de lo que en una imagen hay. Lamentablemente su funcionamiento e instalación son algo enrevesadas y requiere de una configuración que depende mucho de la máquina en la que se trabaje y de la estructura de directorios que esta tenga, pero una vez configurada su funcionamiento es el esperado.

4.2.6 Herramientas de Deep Learning: Caffe

Esta herramienta es usada en nuestro proyecto debido a que la herramienta NeuralTalk tiene dependencia de este proyecto para extraer características de las imágenes y luego poder realizar una predicción con ellas. Además este proyecto está bien estructurado y en si se sigue correctamente su documentación es relativamente fácil de instalar.

4.2.7 Herramientas de programación: MATLAB

Tuvimos que instalar MATLAB porque NeuralTalk usa un script de MATLAB para poder preparar las imágenes para extraerles las características, además se usa el wrapper de caffe para MATLAB.

4.2.8 Herramientas de desarrollo de servidores: Flask

Se ha estudiado entre varias posibles herramientas, entre ellas están:

- Axis2/C
- GSOAP
- Tomcat

Se ha escogido esta herramienta en concreto porque sobre todas las demás su funcionamiento era muy inmediato y además se escribe en Python, que es un idioma muy versátil y fácil de usar. El hecho de que esta herramienta tenga un funcionamiento y programación tan sencilla la hace una herramienta que, a nuestro parecer, destaca sobre el resto y es interesante trabajar con ella. Además tiene una documentación sencilla, repleta de ejemplos y explicaciones para poder realizar tu servidor. Sus ejemplos son muy útiles y funcionan a la primera, sin tener que configurar casi nada. Tiene una forma de establecer los URLs de forma muy sencilla y es fácil estructurar tu servidor con esta API. A pesar de ser una API tan fácil de usar tiene una gran potencia y se pueden construir con ella servidores bastante grandes y fácilmente escalables, por tanto se convierte en la herramienta perfecta para la programación de nuestro servidor.

4.2.9 Manual del Programador

En esta sección se procederá a la explicación detallada de cómo instalar las herramientas necesarias y qué herramientas son necesarias para trabajar sobre este proyecto.



Figura 4.1: Página de descarga del JDK de Java

■ Instalación del JDK

La primera, y más esencial de las herramientas, es el JDK de java, que es el set o conjunto de herramientas y librerías para los desarrolladores de java.

Primero deberemos ir a la página de **Oracle** en la que descargaremos el jdk, la página debería tener un aspecto más o menos como este

En dicha página tendremos que aceptar la licencia y posteriormente descargar el JDK que sirva para nuestra máquina. Una vez hemos descargado dicho archivo, lo ejecutamos. Una vez ejecutado seguimos los siguientes pasos para su instalación.



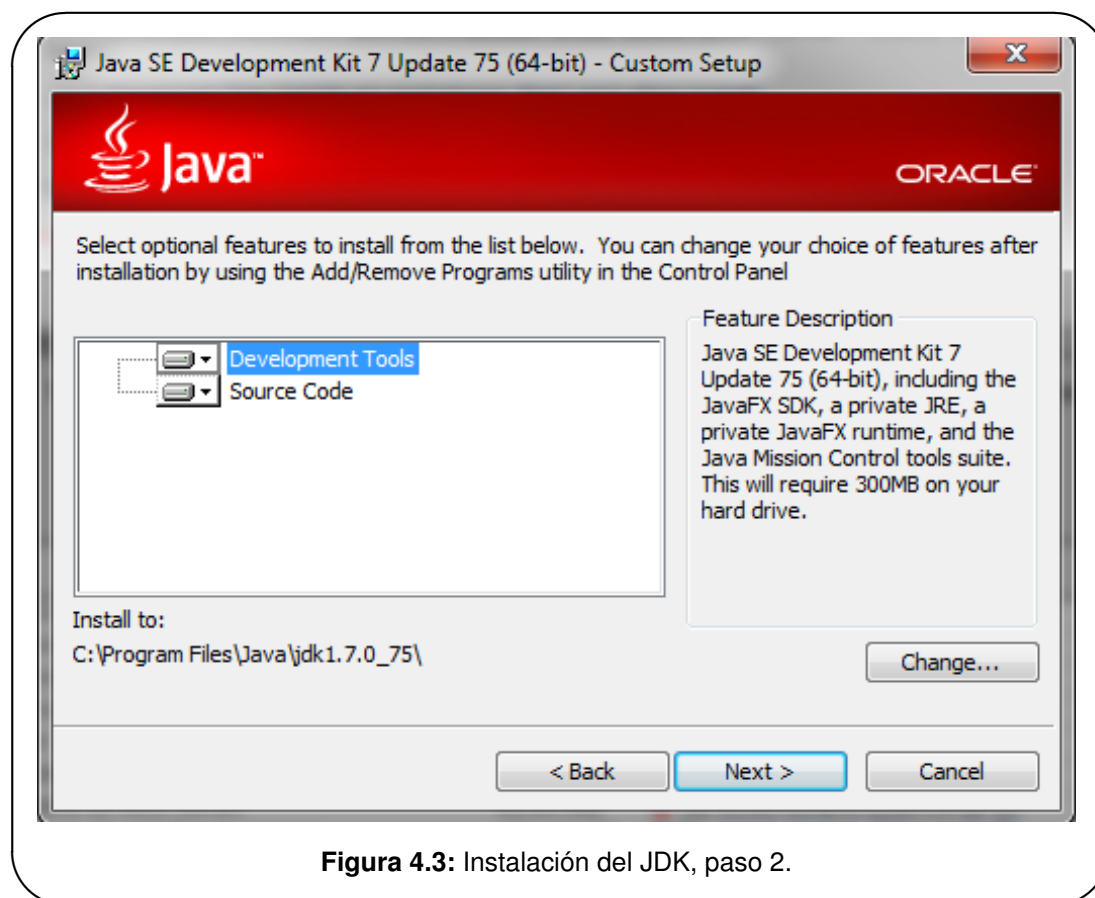




Figura 4.4: Instalación del JDK, paso 3.



Figura 4.5: Instalación del JDK, paso 4.

5. ESTADO DEL ARTE

En este apartado...

6. ASPECTOS RELEVANTES DEL DESARROLLO DEL PROYECTO

En este apartado se introducen los aspectos más relevantes del proyecto.

6.1 Resolviendo puzles en un ordenador

La resolución...

7. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

En este apartado introduciremos en primer lugar las dificultades encontradas durante el desarrollo del proyecto. Posteriormente, se hablará de las conclusiones que se obtienen tras el desarrollo del proyecto. Por último, se expondrán las líneas de trabajo futuras.

7.1 Dificultades encontradas

Durante el desarrollo del proyecto nos hemos encontrado varias dificultades que han hecho que el proyecto se retrase considerablemente y su avance no haya sido ni fácil ni rápido.

7.1.1 Dificultades con GSOAP y Apache

Este fallo es el que más ha retrasado al proyecto y ha supuesto una dificultad enorme a la hora de llevar a cabo el mismo.

Empezamos con que para usar GSOAP se tuvo que estudiar una serie de cosas para poder adquirir los conocimientos necesarios para usar la herramienta, dichos conocimientos serán listados aquí:

- **XML:** Se tuvo que coger un nivel adecuado en el uso de XML ya que la herramienta GSOAP se basa en el uso de este tipo de archivos como medio de comunicación en las distintas peticiones y respuestas que procesa. Además el XML también es necesario para comprender el funcionamiento de SOAP y de WSDL, los cuales son completamente necesarios para entender el funcionamiento de la herramienta GSOAP.
- **SOAP:** Esta especificación se tuvo que estudiar para comprender el funcionamiento de la herramienta GSOAP y en qué se basaba su funcionamiento, entender el por qué debía funcionar la herramienta y como se realiza la comunicación gracias a ella. Aunque el SOAP no es usado directamente cuando usas GSOAP es necesario conocer esta especificación ya que GSOAP sí que usa WSDL, para el cual tenemos que tener un conocimiento básico, al menos, de SOAP para poder usarlo.
- **WSDL:** Esta otra especificación sí que se usa directamente en la herramienta GSOAP y básicamente con ella vertebras toda la aplicación que vas a hacer, de hecho tienes dos opciones:

La primera es usar un archivo WSDL donde especificas las operaciones que el servidor va a realizar, después con la herramienta GSOAP generas todos los stubs y documentos necesarios para hacer tu aplicación.

La segunda sería a través de un documento de tipo .h o una cabecera de C. Con el cuál también generas los stubs y documentos necesarios para programar tu servidor, entre dichos documentos se encontrará un archivo WSDL que contendrá la especificación de las operaciones que hay dentro del fichero cabecera que hayas usado. Pero incluso en esta opción necesitas entender SOAP y WSDL porque a través de comentarios tienes que especificar características que irán directamente al fichero WSDL, y que serán necesarios para el correcto funcionamiento del servidor.

Una vez se ha estudiado lo anterior se paso al estudio de la documentación de la herramienta GSOAP, además de el intento de hacer que funcionen sus ejemplos. Cuando se consiguió que funcionarían sus ejemplos se paso a la programación de un servidor propio, una vez se programo y se hicieron las pruebas de que estaba bien programado se procedió a intentar que este funcionará desde un cliente GSOAP. Para que funcionará con el cliente GSOAP se hizo una investigación de cómo hacer que el servidor funcionará en localhost y, siguiendo la recomendación que en la documentación de GSOAP encontramos, se instaló Apache y se inteto usar el módulo de Apache para su funcionamiento con GSOAP.

Como conclusión sacamos que, tras una larga investigación y mucho tiempo dedicada a esta herramienta, esta herramienta no tenia la documentación suficiente como para poder hacerla funcionar con Apache y, a pesar de haberlo intentado muchas veces, no conseguimos que el servidor GSOAP programado por nosotros devolviera alguna vez un resultado coherente al cliente. Finalmente desechamos la opción de trabajar con esta herramienta y le dimos un giro al proyecto con el que esperábamos tener avances más rápidos y mejores, optamos por la programación de un servidor en Flask.

7.1.2 Dificultades con DeepBeliefSDK

Al final no se pudo usar.

7.1.3 Dificultades en la instalación de Herramientas

Caffe tuvimos problemas de compatibilidad con open-cv Neuraltalk tuvimos problemas con la cantidad de dependencias que tenía.

7.1.4 Dificultades con NeuralTalk

Tuvimos problemas con la herramienta de MATLAB al ejecutar un script desde el servidor.

7.1.5 Dificultades con Android

Problemas al conectarnos con el servidor.

7.2 Conclusiones del Proyecto

7.3 Lineas de trabajo Futuras

