Especificación de Requerimientos del Sistema

para

through•Code

Versión 1.0

Preparada por Christian Eduardo Rodríguez Palacios

DaGeeks

22 de octubre de 2014

Contenido

Contenido ii

Revisiones 3

1 Introducción 3

1.1 Alcance del producto 3

1.2 Definiciones, acrónimos y abreviaciones 3

1.3 Referencias 4

1.4 Resumen del documento 4

2 Descripción General 5

2.1 Características generales del producto 5

2.2 Ambiente de operación 6

2.3 Limitaciones de diseño e implementación 6

3 Requerimientos 7

3.1 Conexión directa con robot 7

3.2 Interfaz drag-n-drop 7

3.3 Instrucciones parametrizadas y sincronizadas 7

3.4 Permanencia de los programas 7

4 Requerimientos de Interfaces Externas 8

4.1 Interfaces de usuario 8

4.2 Interfaces de hardware 8

4.3 Interfaces de software 9

4.4 Interfaces de comunicación 9

5 Atributos de calidad 10

6 Otros Requerimientos 10

Revisiones

| Versión | Autor | Descripción de la Versión | Fecha completada |
| --- | --- | --- | --- |
| 1.0 | Christian Eduardo Rodríguez Palacios | Versión Inicial. | 24/10/2014 |

# Introducción

El proyecto throughCode busca que más personas estén interesadas en el mundo de la programación, al ofrecerles una forma fácil e intuitiva de crear programas para controlar un robot, utilizando su Smartphone con una App.

En este documento, se describen las características del producto a desarrollar, junto con los requerimientos de operación y funcionalidad incluida en la aplicación, así como las restricciones en cuanto a tecnología y capacidades se refiere.

## Alcance del producto

El propósito de este proyecto es generar una aplicación que acerque a las personas a la experiencia de programar de forma que no requieran tener conocimientos avanzados sobre la operación de un robot, únicamente sabiendo los principios y funcionamiento básicos de uso de un aparato electrónico de uso cotidiano, como lo es una computadora, un Smartphone o una tableta.

El beneficio que busca genera la aplicación es promover el interés de las personas en la programación, ya sea de hardware o software, a la vez que conocen el fundamento de operación de muchos de los dispositivos utilizados en la actualidad y que mueven el mundo.

## Definiciones, acrónimos y abreviaciones

API: Application programming interface, en español interfaz de programación de aplicaciones. Es un formato de lenguaje y mensajería utilizado por una aplicación para comunicarse con el sistema operativo y otros programas, mediante llamadas de función.

Bluetooth: Tecnología de cotro alcance (10 metros) que permite a computadotras y dispositivos móviles conectarse entre ellos de forma inalámbrica.

Dispositivo móvil: Computadoras pequeñas que son capaces de transportarse en un bolsillo.

Dpi: Dots per inch, en español Puntos por pulgada. Es un número que indica la cantidad de que habrá por pulgada de pantalla, de modo que un valor más alto indica una mayor resolución.

IDE: Software que sirve de apoyo para la escritura de código fuente, proporcionando herramientas que faciliten su desarrollo.

SQLite: Es una librería que implementa un motor de base de datos SQL sin necesidad de utilizar un servidor, y que forma parte del código de la aplicación.

## Referencias

https://source.android.com/source/code-style.html

http://www.alegsa.com.ar/Dic/dispositivo%20movil.php

http://www.mhe.es/universidad/informatica/8448136640/archivos/apendice\_general\_4.pdf

http://support.apple.com/kb/HT3038

http://www.fmedda.com/es/article/dpi\_ppi

http://www.pcmag.com/encyclopedia/term/37856/api

http://www.sqlite.org/about.html

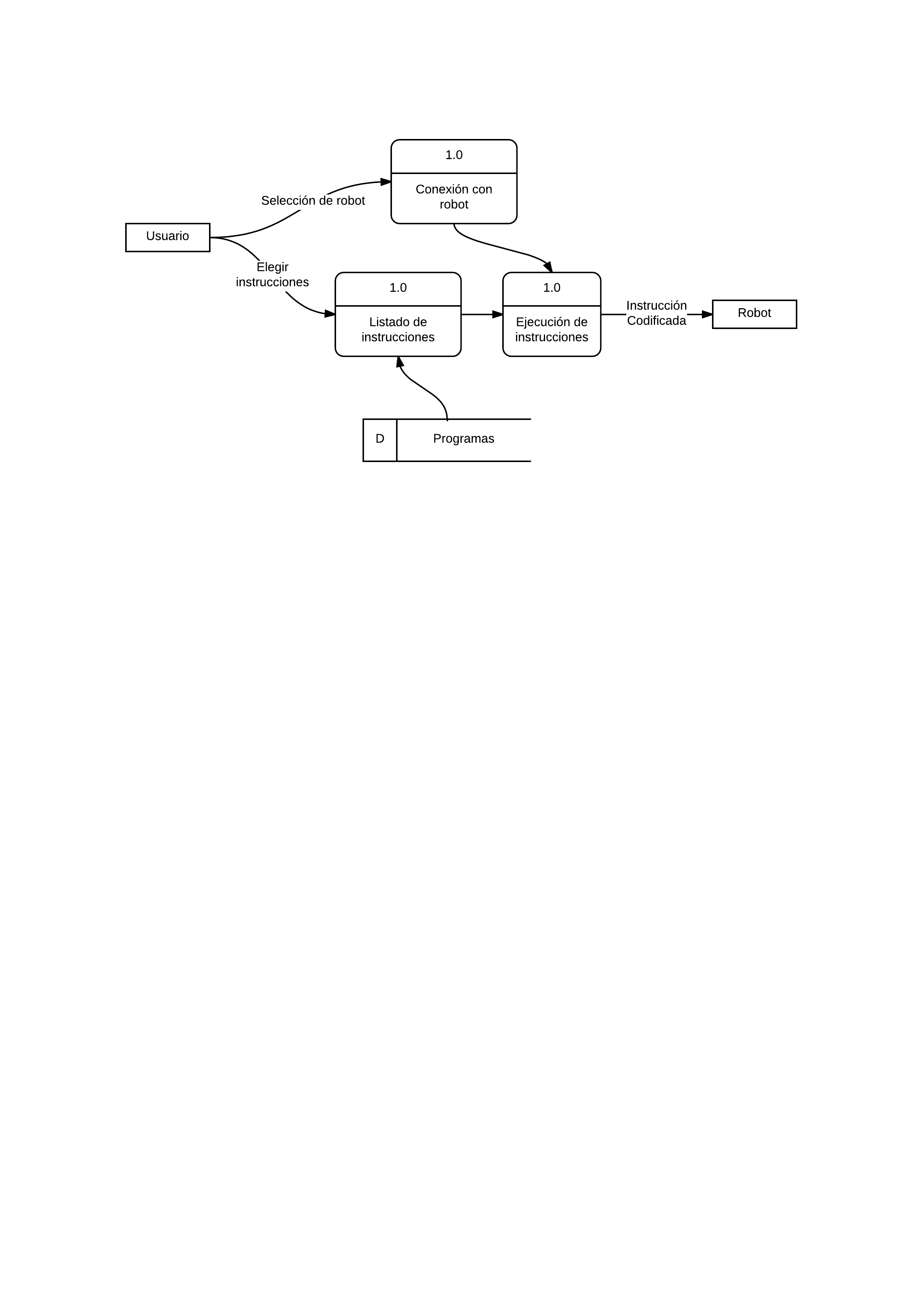
## Resumen del documento

Se incluye una sección con las características que el producto final ha de tener incorporadas, junto con el ambiente de operación utilizado como base para el desarrollo de la aplicación. Posteriormente, se incluyen consideraciones y reservas que tiene el proyecto que limitan las opciones de implementación.

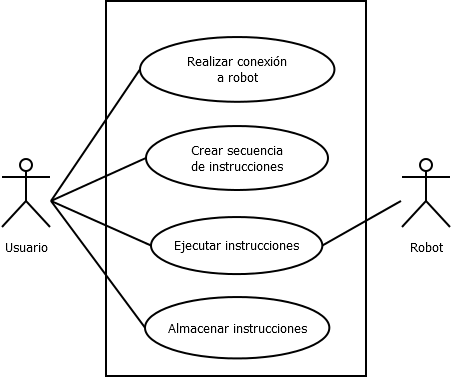
Se listan los requerimientos funcionales de la aplicación y una breve descripción de cada uno, junto con los requerimientos externos a la aplicación y que deben ser utilizados como línea base para definir las capacidades e implementación de las características.

# Descripción General

## Características generales del producto

* La aplicación generará programas mediante instrucciones predefinidas, que el usuario podrá usar para crear sus programas.
* Cada instrucción puede ser configurada por el usuario antes de su ejecución.
* Las instrucciones que forman un programa se presentarán en una lista, mostrando sus parámetros de ejecución y el estatus de su ejecución en el robot.
* Al iniciar, la aplicación buscará robots con los que pueda interactuar y se conectará con ellos antes de iniciar cualquier operación.
* Los programas podrán almacenarse dentro del dispositivo móvil para poder ejecutarse, modificarse o eliminarse según el usuario lo considere.

### Modelo de casos de uso



## Ambiente de operación

La plataforma de hardware para la aplicación será el Smartphone, de gama media-baja. La resolución de pantalla será de 160 dpi (resolución media), y el dispositivo deberá tener incorporada la tecnología Bluetooth para comunicarse con el robot.

La aplicación estará diseñada considerando que operará en el sistema operativo Android, en su versión 4.4 (KitKat, nivel de API 19). La versión mínima del sistema operativo que soportará la aplicación será 4.0 (Ice Cream Sandwich, nivel de API 14). No estarán corriendo aplicaciones en segundo plano ni servicios de terceros que afecten el rendimiento de las aplicaciones.

## Limitaciones de diseño e implementación

La aplicación será programada de forma nativa para el sistema operativo Android, por lo que el lenguaje de programación de facto será Java, utilizando el IDE Eclipse en su versión para desarrollo de aplicaciones de Android.

La aplicación solo podrá conectarse con los robots únicamente utilizando la tecnología de Bluetooth, y no hará uso de otros protocolos inalámbricos. Así mismo, la sincronización inicial entre el dispositivo móvil y el robot será manual, y al terminar la ejecución de la aplicación, la conexión se terminará. Las instrucciones serán enviadas al robot utilizando el protocolo de codificación predefinido por el proyecto.

La aplicación deberá operar de forma fluida y dinámica en su interacción, no haciendo esperar al usuario más de 0.5 segundos al interactuar con los elementos de la interfaz gráfica. Sólo podrá hacerlo esperar cuando la aplicación realice operaciones en conjunto con el robot, ya sea al comunicarse o ejecutar instrucciones en él, en cuyo caso deberá informar al usuario el estado de la conexión y la duración de estos tiempos de espera no debe superar los 10 segundos.

Se usará SQLite para almacenar la información de ejecución de los programas. El tamaño de los programas, es decir, la cantidad de instrucciones que pueden tener almacenadas está limitada a 50 instrucciones por programa.

# Requerimientos

## Conexión directa con robot

Al inicio de la ejecución de la aplicación, el robot deberá utilizar las capacidades de Bluetooth del dispositivo móvil para detectar robots con los que pueda emparejarse para enviar y ejecutar las instrucciones. En la pantalla inicial de la aplicación, se deberá listar los robots con los que puede realizar la conexión, de modo que el usuario pueda seleccionar el robot con el cual trabajar.

## Interfaz drag-n-drop

La interfaz para definir las reglas de los programas deberá componerse de tres elementos principales, una lista de las instrucciones que la aplicación puede enviar al robot para su ejecución (el pool de instrucciones), otra lista con la secuencia de instrucciones que conforman el programa que se ejecutará en el robot (el stack) y una barra con las opciones relacionadas con ejecución y almacenamiento de programas.

El usuario podrá arrastrar instrucciones desde el pool de instrucciones hacia el stack. El listado en el stack mostrará en cada renglón la instrucción y los parámetros de ejecución.

## Instrucciones parametrizadas y sincronizadas

Cada instrucción puede configurarse según considere el usuario, de modo que cada una se comporte de acuerdo a estos parámetros.

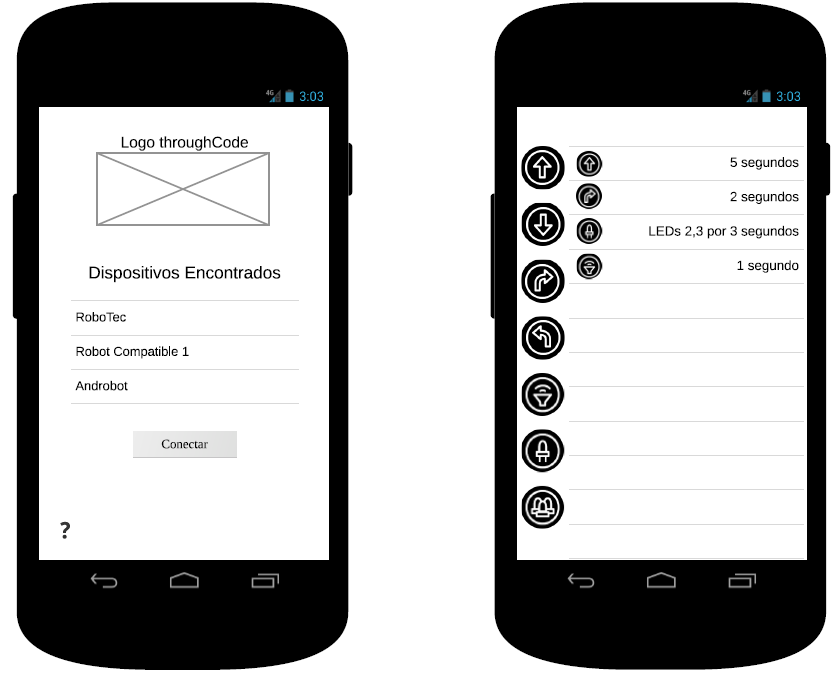
Las instrucciones serán capaces de reportar la situación de su ejecución en el robot, dependiendo si no se ha iniciado la ejecución, se están ejecutando, se ejecutaron con errores u operaron satisfactoriamente.

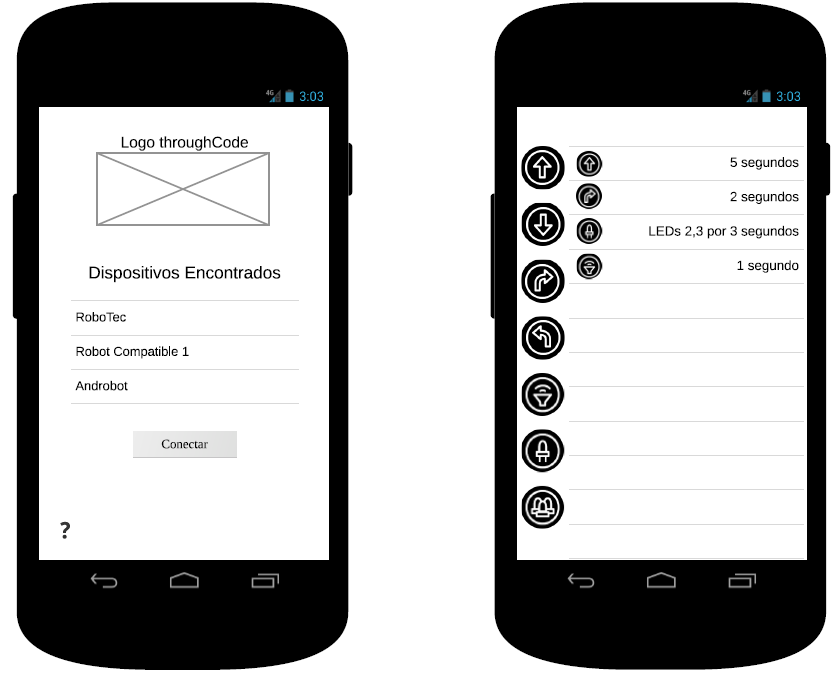
## Permanencia de los programas

Los programas que el usuario genere podrán almacenarse dentro del dispositivo móvil. Los programas que ya hayan sido almacenados también podrán cargarse

# Requerimientos de Interfaces Externas

## Interfaces de usuario

La interfaz gráfica se compone de dos pantallas. La primera pantalla será utilizada para iniciar la conexión con un robot que sea compatible con el protocolo. Mostrará el logo de la aplicación, una lista con los robots que el dispositivo móvil ha encontrado, un botón para iniciar la conexión y un botón de ayuda.

La segunda pantalla mostrará las instrucciones que pueden enviarse al robot para su ejecución, una lista con las instrucciones que conforman el programa actual junto con sus parámetros de ejecución y el estado de su ejecución. En la parte superior, la aplicación mostrará las opciones para ejecutar el programa, guardarlo, abrir un programa existente o eliminar los creados anteriormente.

## Interfaces de hardware

El hardware utilizado para la conexión con el robot será la antena de Bluetooth del dispositivo móvil, únicamente pudiendo conectarse mediante esta tecnología con solo un robot. La comunicación entre ambos dispositivos será mediante un protocolo de comunicación definido previamente, el cual no forma parte de los alcances de este proyecto.

Los robots a los que se podrá conectar deberán ser compatibles con los protocolos de comunicación predefinidos y ser capaces de comunicarse vía Bluetooth.

La aplicación hará uso de los gestos en la pantalla táctil.

## Interfaces de software

La aplicación estará diseñada para el sistema operativo Android 4.4 (KitKat), aunque soportará como versión mínima la versión 4.0 (Ice Cream Sandwich).

La base de datos utilizada para almacenar programas será SQLite, dentro del dispositivo móvil.

Los datos de entrada que recibe la aplicación son las instrucciones que deberá enviar al robot, junto con los parámetros que estas requieran y hayan sido asignadas por el usuario. También recibirá mensajes del robot sobre el estatus de la ejecución de la instrucción actual. Como datos de salida, enviara instrucciones codificadas al robot con los parámetros respectivos y mostrará el estatus de la ejecución de cada instrucción al usuario.

En caso de que el sistema disponga de niveles de energía críticos (debajo del 3%) o cuando el dispositivo móvil se suspenda, la aplicación deberá cerrar el canal de comunicación con el robot y terminar su ejecución.

## Interfaces de comunicación

La comunicación utilizada entre el dispositivo móvil y el robot será mediante un protocolo de comunicación que codificará las instrucciones y sus parámetros de ejecución para que el robot las interpreta y las lleve a cabo, así como respuestas que enviará el robot con respecto al estatus de éste y las instrucciones que ha recibido. Este protocolo no estará encriptado, únicamente con la protección que ofrece Bluetooth.

# Atributos de calidad

La aplicación sólo usará la antena de Bluetooth del dispositivo móvil para comunicarse con el robot. Este canal de comunicación estará disponible sólo cuando la aplicación sea utilizada por el usuario. En otro caso, el canal debe cerrarse inmediatamente. No podrá interactuar con otro tipo de dispositivos.

La aplicación ocupará un máximo de 20 MB para su instalación. Así mismo, no utilizará más del 30% del CPU del dispositivo móvil de acuerdo al ambiente de operación definido.

La aplicación sólo será compatible con aquellos robots que sean compatibles con el protocolo utilizado en el proyecto, y se comunicará con ellos únicamente por Bluetooth.

# Otros Requerimientos

El programa estará en español.

La aplicación se regirá por la licencia de código libre GNU GPL versión 3.

El código se apegará a los lineamientos de la guía de estilo para desarrollos en Android.