



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Aplicaciones Informáticas

Grupo

Cavaiuolo Adrián Gabriel

2do cuatrimestre 2020

Fecha de entrega: 08/03/2021

Índice

| | |
|---|-----------|
| Índice | 1 |
| Lineamientos generales opción 1 | 3 |
| Objetivo del Desarrollo | 4 |
| Diseño | 5 |
| Requerimientos Previos | 5 |
| Pasos del Proceso | 5 |
| Modelo de Proceso Inicial | 6 |
| Modelo de Proceso Conceptual a Futuro | 6 |
| Flujo de Uso de la Aplicación | 6 |
| Paso | 8 |
| Plataforma a Utilizar | 9 |
| Xamarin.Forms | 9 |
| Arquitectura de Xamarin.Forms | 9 |
| Text To Speech | 9 |
| Comandos de Voz | 10 |
| Avanzar | 10 |
| Repetir | 10 |
| Volver | 10 |
| Terminar | 10 |
| Buscar | 10 |
| Cerrar | 10 |
| Alternativas Tecnológicas de Reconocimiento de Voz | 11 |
| Reconocimiento de voz con el servicio de voz de Azure | 11 |
| Reconocimiento por Android SpeechRecognizer | 12 |
| ACR Speech Recognition / Shiny | 12 |
| Comparativa de Tecnologías | 12 |
| Análisis de Diego Martín Costa | 13 |
| Repositorio Principal | 13 |
| Casos de Prueba | 14 |
| Arquitectura de la Solución | 18 |
| Diseño Básico | 18 |
| Paquetes | 18 |
| Temporizadores (desarrollo futuro) | 19 |
| Video Demostrativo | 19 |

| | |
|--------------------|-----------|
| Repositorio | 20 |
| Hosting | 20 |

Lineamientos generales opción 1

Agenda procedimental interactiva con tecnología de voz.

Integrar una app que por voz interactúe con un archivo, realizando la búsqueda en particular de un contenido procedimental.

Obtenido el contenido envíe el mismo en modo texto, el cual se pueda reproducir en audio interactivo en función de los pasos que representa el texto cargado para una determinada acción y sus parámetros temporales indicados en el registro.

Interacción del usuario confirmando que completo el paso por voz y por la aplicación ante la consulta disparada por el parámetro temporal del guión correspondiente al cumplimiento del paso.

Observar la posibilidad de dependencia de pasos además de la temporalidad, antes de iniciar un nuevo paso.

Sumar para cada paso un descriptor de elementos necesarios para su realización de corresponder, sumar un atributo de características especiales para la ejecución del paso

En particular en documento [Control de dispositivos por comandos de voz \(Diego Costa\)](#)

PARTE II

Análisis de Soluciones de reconocimiento de voz, elección de la solución y fundamentación.

Dentro de las soluciones ver Kaldi/Vosk

- <http://kaldi-asr.org/>
- <https://alphacephei.com/vosk/>

Objetivo del Desarrollo

En base a lo planteado en los lineamientos, lo que se propone es el desarrollo de una plataforma base que sirva de modelo o de arquitectura base para la creación de aplicaciones que se basen en el seguimiento de pasos de un proceso para llegar a un objetivo.

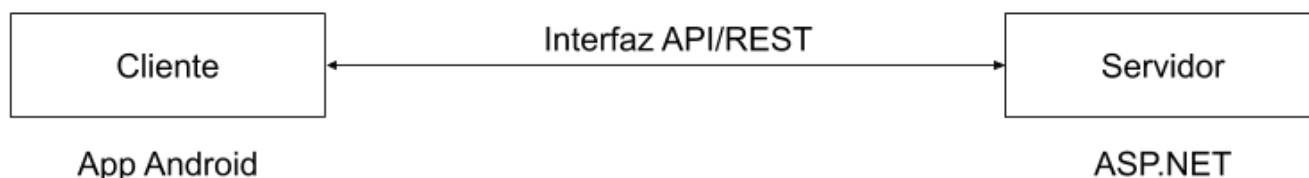
Es decir, en lugar de desarrollar una aplicación concreta para el seguimiento de tareas específicas como “recetas de cocina” o “ensamblaje de una jaula para loros”, se propone la creación de una aplicación genérica que cualquier interesado pueda utilizar para suplir sus necesidades. De esa forma, una empresa que fabrica un producto que requiere ensamblaje por el comprador, podría tener su propia implementación de esta aplicación, configurada y adaptada para su negocio en particular.

Modelos similares pueden encontrarse con aplicaciones comunes como “e-commerce”, que ofrecen un servicio tipo **Software as a Service (SaaS)**, en el cual un vendedor o proveedor de algún bien puede crear su propio sitio de ventas online, utilizando una plataforma desarrollada de forma genérica para este fin. Ejemplos de este modelo pueden ser: TiendaNube, EmpreTienda, u otros ejemplos de modelo similar como Wix.

Si bien esta modalidad no sería exactamente igual, ya que no pretendemos cobrar por ningún servicio, al tratarse de una aplicación open source, mantendríamos la idea de la creación de una aplicación base customizable y reutilizable distintos modelos de negocio, esta vez aplicada a la realización de procesos por el usuario final.

Diseño

El diseño básico sería la generación de dos aplicaciones, una aplicación **Servidor** para la generación, configuración y exposición (a través de una interfaz) de los procesos que se pueden consultar y generar, y una aplicación **Cliente** en principio Android que consultará, descargaría y permitiría ejecutar y seguir los pasos de los procesos creados.



A partir de este modelo simple, la creación de las distintas implementaciones de procesos dependerá de quien tenga interés en su implementación. Una persona podría, sin saberlo, tener en su teléfono varias implementaciones de la misma aplicación, todas con distintos objetivos. Por ejemplo:

- Procesos de preparación de platos culinarios.
- Procesos de ensamblaje de asientos y escritorios.
- Procesos de trámites a realizar de forma remota en algún sistema del estado.

Siempre partiendo de una base común de que para realizar un proceso, se tienen que tener ciertas cuestiones en cuenta:

Requerimientos Previos

Todos los procesos pueden tener requerimientos previos. Estos pueden ser, por ejemplo:

- Las herramientas necesarias para la construcción de una pajarera.
- Los ingredientes necesarios para llevar a cabo una receta.
- Los papeles necesarios para el inicio de un trámite.

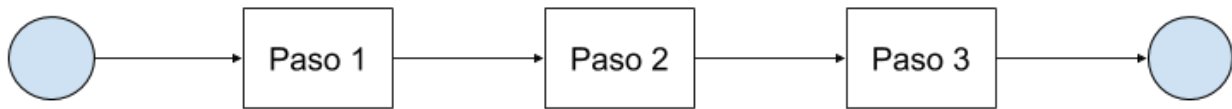
Por tanto, los procesos deben comenzar con un listado de los requerimientos que se deben cumplir a la hora de empezar el proceso.

Pasos del Proceso

Una vez los requerimientos previos del usuario han sido cumplidos, el proceso se inicia con el primer paso. En principio, la ejecución de pasos, se llevará a cabo de forma lineal, con la posibilidad para el usuario de volver a atrás (en caso de que sea necesario, para refrescar un concepto, o idea), o bien, la posibilidad de repetir una instrucción.

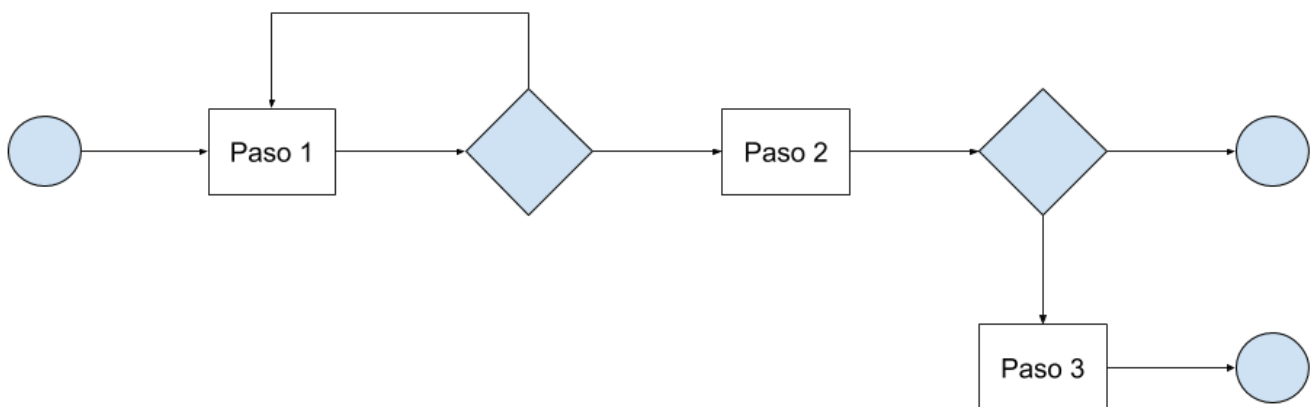
Para permitir el avance o retroceso del usuario a través del proceso elegido, se contarán con determinados comandos de voz que la aplicación podrá interpretar y actuar en consecuencia. Ver [Comandos de Voz](#).

Modelo de Proceso Inicial



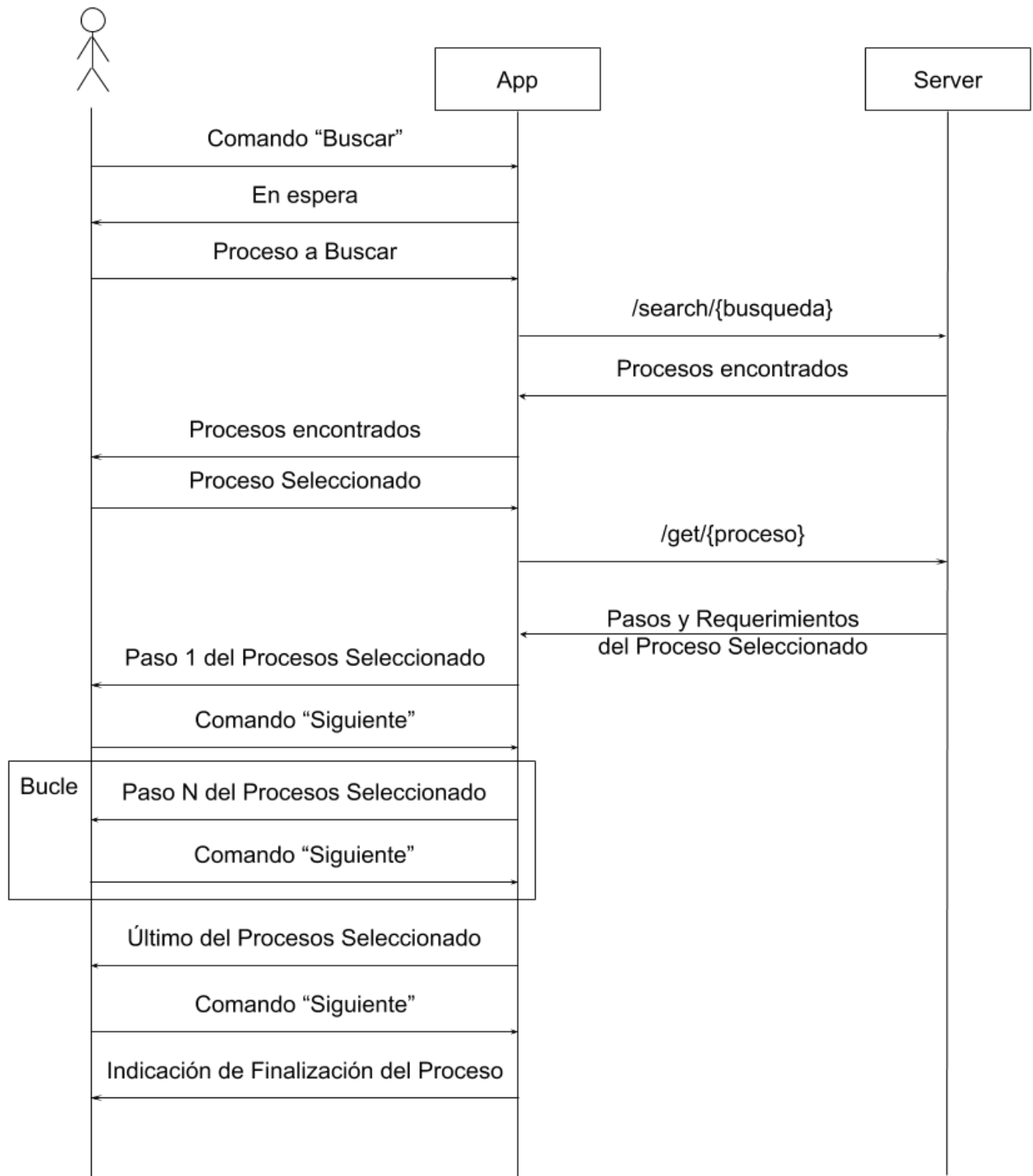
Modelo de Proceso Conceptual a Futuro

Más adelante, con el debido tiempo de desarrollo posterior, podrían incluirse nuevos conceptos como condicionales o bucles, siempre manteniendo la compatibilidad.



Flujo de Uso de la Aplicación

La idea del flujo de uso de la aplicación, es que pueda ser controlada íntegramente por voz. Desde el momento en que se abre la aplicación, hasta que se cierra.



Paso

Un paso es una actividad o tarea que el usuario debe realizar dentro de un proceso. La información de cada paso estará almacenada en el servidor, y descargada por el usuario cuando desee realizar el proceso.

Cada paso deberá contar con un nombre identificador, una descripción textual que será leída e interpretada por la aplicación y convertida a audio para el usuario (ver [Text To Speech](#)). Además, se podrá cargar una imagen referencial, de ser necesaria.

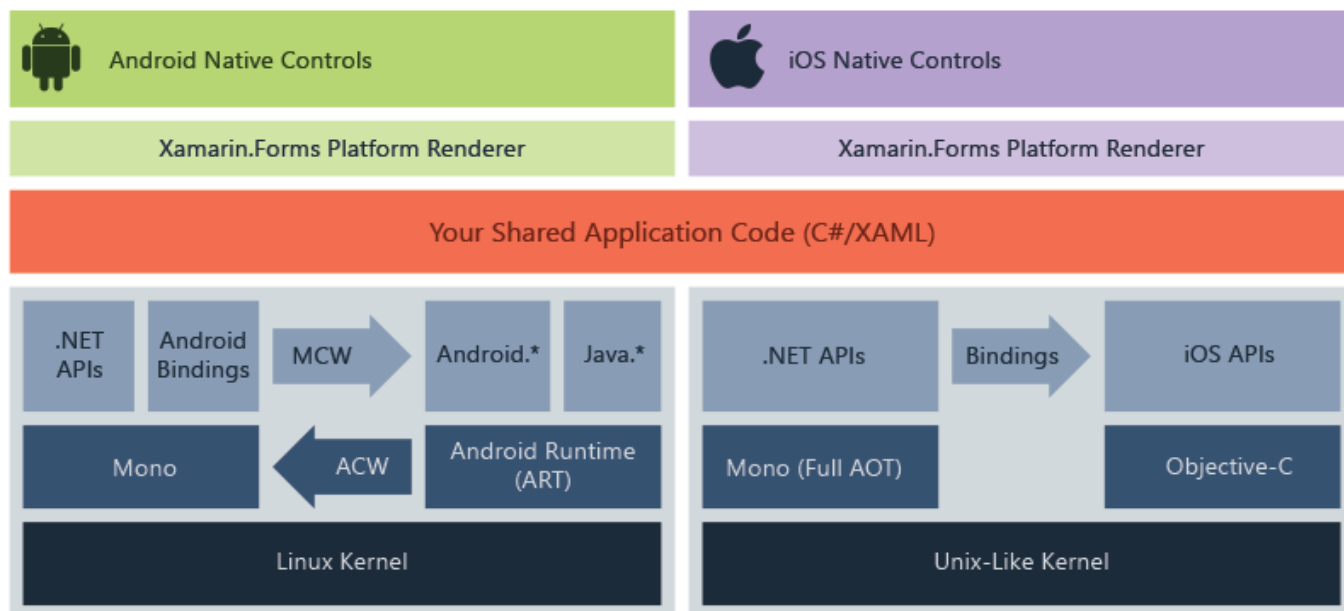
Plataforma a Utilizar

Xamarin.Forms

Xamarin.Forms permite compilar aplicaciones nativas mediante un kit de herramientas de interfaz de usuario multiplataforma de .NET para factores de forma correspondientes a móviles, tabletas y equipos de escritorio en Android, iOS y más.

Arquitectura de Xamarin.Forms

Xamarin.Forms proporciona una API coherente para crear elementos de interfaz de usuario entre plataformas. Esta API se puede implementar en **XAML** o **C#**, y admite el enlace de datos para patrones como Model-View-ViewModel (MVVM).



<https://docs.microsoft.com/es-es/xamarin/get-started/what-is-xamarin-forms>

Text To Speech

Xamarin cuenta con una implementación nativa y multiplataforma para transformar texto en audio. Esta será utilizada para leer al usuario la descripción de los pasos a realizar, así como los mensajes de confirmación de comandos que sean necesarios.

<https://docs.microsoft.com/en-us/xamarin/essentials/text-to-speech>

Comandos de Voz

Debido a la linealidad descrita previamente para los procesos, se contará con tres opciones de comandos, que podrán tener varias formas de ejecutarse. Probablemente las opciones evaluadas para cada comando dependerán de la calidad de interpretación de la librería de reconocimiento de voz utilizada.

Se apunta a que todos los comandos sean cortos y concretos, de forma de minimizar los errores de comprensión de la aplicación.

Avanzar

Comando básico para pasar el siguiente paso del proceso. Se evalúan las siguientes alternativas: “siguiente”, “listo”, “próximo”, “avanzar” y/o sinónimos de las mismas.

Repetir

Comando utilizado para repetir las instrucciones del paso actual, en caso de que el usuario no lo haya escuchado bien, o estaba ocupado o distraído. Se evalúa: “repetir”, “de vuelta”

Volver

Comando para volver a atrás un paso que quizás fue confirmado por accidente, o erróneamente. Se evalúan: “volver”, “retroceder”, “anterior”.

Terminar

Este comando apunta a una terminación del proceso independientemente del punto en el flujo en el que se encuentra el usuario. Se evalúan: “terminar” y “finalizar”. Deberá requerir una confirmación, ya que de dispararse por error, estaría perjudicando al usuario.

Buscar

Este comando se utilizará fuera de los flujos estándar, justamente para la búsqueda del proceso a realizar. No sería del todo necesario, ya que se entiende que el usuario no estaría “ocupado” antes de iniciar el proceso, pero mantendría una coherencia a lo largo de la aplicación, siendo todos los demás comandos activados por voz.

Cerrar

Este comando cierra completamente la aplicación. Deberá requerir una confirmación, ya que de dispararse por error, estaría perjudicando al usuario.

Alternativas Tecnológicas de Reconocimiento de Voz

Ante todo, se debe aclarar que la principal característica que se observará dadas las distintas tecnologías de reconocimiento de voz accesibles es (además obviamente de que se encuentre disponible para la tecnología definida) su gratuidad o las prestaciones que ofrezca en su versión gratuita.

También se tomarán en cuenta otros factores:

- Si es open source, ya que el acceso al código podría ayudar a resolver problemas.
- Si funciona sin necesidad de un acceso a internet, ya que podrían utilizarse datos móviles para la descarga del proceso, y luego continuar desconectado de la red.
- Si es multiplataforma. Aunque inicialmente apuntamos a una aplicación Android, la posibilidad de escalar a otras plataformas, será considerada.

Reconocimiento de voz con el servicio de voz de Azure

Dada la plataforma que determinamos utilizar para el desarrollo, la librería más recomendable a utilizar sería la de Azure.

<https://docs.microsoft.com/es-es/xamarin/xamarin-forms/data-cloud/azure-cognitive-services/speech-recognition>

Esta expone una interfaz única tanto para iOS como para Android, lo cual es un punto a favor. Sin embargo, no es completamente gratuita, y dependiendo de su uso, tiene licenciamiento:

<https://azure.microsoft.com/es-es/pricing/details/cognitive-services/speech-services/>

En principio, la versión gratuita podría servir para desarrollo:

| | | |
|-------------------|---|--|
| Speech to Text | Estándar | 5 horas de audio gratis al mes |
| | Personalizado | 5 horas de audio gratis al mes Hospedaje de puntos de conexión: 1 modelos gratis al mes ² |
| | Audio multicanal de transcripción de conversaciones <small>VERSIÓN PRELIMINAR 3</small> | 5 horas de audio gratis al mes |
| Text to Speech | Estándar | 5 million caracteres gratis al mes |
| | Neuronal | 0.5 million caracteres gratis al mes |
| | Personalizado | 5 million caracteres gratis al mes Hospedaje de puntos de conexión: 1 modelos gratis al mes |
| Traducción de voz | Estándar | 5 horas de audio gratis al mes |

| | | |
|---------------------|-----------------------------|-------------------------------------|
| Speaker Recognition | Comprobación del hablante | 10,000 transacciones gratis por mes |
| | Identificación del hablante | 10,000 transacciones gratis por mes |

Pero consideramos que es innecesario partir con estas limitaciones cuando hay otras alternativas gratuitas.

Reconocimiento por Android SpeechRecognizer

El reconocimiento de voz por Android SpeechRecognizer es gratuito y tiene una versión instalable desde **Xamarin**. No es multiplataforma.

<https://developer.android.com/reference/android/speech/SpeechRecognizer>

ACR Speech Recognition / Shiny

ACR es una librería que actualmente se encuentra integrada al framework Shiny. Se consideró su utilización, por las prestaciones que ofrece (multiplataforma, gratuito, open source), pero parecería requerir la utilización completa del framework para el desarrollo de la aplicación, y no parece ser posible la utilización de sólo esta característica.

<https://github.com/aritchie/speechrecognition>

<https://github.com/shinyorg/shiny>

Comparativa de Tecnologías

| | Azure | Android SDK | ACR |
|-----------------------------------|---|-------------------|-----|
| Multiplataforma | Sí | No (sólo Android) | Sí |
| Gratuito | No (aunque tiene una versión gratuita). | Sí | Sí |
| Funciona sin conexión a internet? | No | No | Sí |
| Es Open Source? | No | No | Sí |

La determinación final de la tecnología a utilizar se cerrará durante el desarrollo de la aplicación, ya que las pruebas y el análisis previo no fueron suficientes para determinar definitivamente la utilización de ninguna en particular.

Análisis de Diego Martín Costa

Basados en el análisis provisto por Diego Martín Costa,

https://github.com/adri-kriptok/tpai7539/blob/master/docs/TP_DiegoCosta.pdf

Repositorio Principal

<https://github.com/adri-kriptok/tpai7539>

Casos de Prueba

| | |
|---------------------------------|---|
| Nombre | TC01 - Inicio |
| Escenario | Al abrir la aplicación, ésta realiza un saludo amistoso al usuario. Invitándolo a hablar e interactuar de ese modo. |
| Pasos para el test | 1. Iniciar la aplicación 2. Esperar hasta que la aplicación cargue, y se termine de escuchar el mensaje. |
| Datos para el test | Tener el volumen del teléfono en un nivel de mediano a alto para poder entender lo que dice. |
| Resultado Esperado | La aplicación emite un un mensaje hardcodeado en la aplicación, por la librería de TextToSpeech , y luego se queda esperando a recibir comandos de habla por el usuario. |
| Resultado Obtenido | La aplicación inicia, y el mensaje se escucha correctamente. |
| Resultado del Test (Pasó/Falló) | Pasó |

| | |
|--------------------|--|
| Nombre | TC02 - Comandos |
| Escenario | Luego del saludo inicial, la aplicación se queda esperando que se exprese un comando. |
| Pasos para el test | 1. Iniciar la aplicación 2. Esperar a que la pantalla de bienvenida termine de inicializar la aplicación. 3. Pronunciar un comando. |
| Datos para el test | Los comandos aceptados en esta instancia pueden ser: <ul style="list-style-type: none"> • Salir • Comandos/Ayuda • Buscar Cualquier otro mensaje recibido, se informará de la incapacidad de entender el mensaje. |
| Resultado Esperado | El resultado esperado, depende del comando. Ver casos: TC03 - Comando "Salir" TC04 - Comando "Comandos"/"Ayuda" TC05 - Comando "Buscar" |
| Resultado Obtenido | N/A |

| | |
|------------------------------------|------|
| Resultado del Test (Pasó/Falló) | Pasó |
|------------------------------------|------|

| | |
|------------------------------------|--|
| Nombre | TC03 - Comando “Salir” |
| Escenario | Luego del saludo inicial, la aplicación se queda esperando que se exprese un comando. El usuario quiere salir de la aplicación. |
| Pasos para el test | 1. Iniciar la aplicación 2. Esperar a que la pantalla de bienvenida termine de inicializar la aplicación. 3. Pronunciar el comando “salir” |
| Datos para el test | N/A |
| Resultado Esperado | La aplicación se despide con un mensaje amistoso, y se cierra la aplicación. |
| Resultado Obtenido | El mensaje se escucha, y la aplicación se cierra correctamente. |
| Resultado del Test (Pasó/Falló) | Pasó |

| | |
|------------------------------------|---|
| Nombre | TC04 - Comando “Comandos”/”Ayuda” |
| Escenario | Luego del saludo inicial, la aplicación se queda esperando que se exprese un comando. El usuario no conoce los comandos que la aplicación entiende, o que se pueden usar en ese momento. |
| Pasos para el test | 1. Iniciar la aplicación 2. Esperar a que la pantalla de bienvenida termine de inicializar la aplicación. 3. Pronunciar cualquiera de los comandos “comandos” o “ayuda”. |
| Datos para el test | Los comandos aceptados en esta instancia pueden ser: <ul style="list-style-type: none"> • Salir • Buscar |
| Resultado Esperado | La aplicación informará de los comandos disponibles. |
| Resultado Obtenido | La aplicación cita los comandos correctamente. |
| Resultado del Test (Pasó/Falló) | Pasó |

| | |
|---------------------------------|--|
| Nombre | TC05 - Comando “Buscar” |
| Escenario | Luego del saludo inicial, la aplicación se queda esperando que se exprese un comando. El usuario quiere buscar un proceso para realizar. |
| Pasos para el test | 1. Iniciar la aplicación 2. Esperar a que la pantalla de bienvenida termine de inicializar la aplicación. 3. Pronunciar el comando “buscar”. |
| Datos para el test | N/A |
| Resultado Esperado | La aplicación entiende el comando, y abre la pantalla del “buscador”. |
| Resultado Obtenido | La aplicación abre el buscador correctamente. |
| Resultado del Test (Pasó/Falló) | Pasó |

| | |
|---------------------------------|---|
| Nombre | TC06 - Comando “Volver” |
| Escenario | Luego de mostrarse la pantalla del buscador, el usuario quiere volver a la pantalla principal. |
| Pasos para el test | 1. Iniciar la aplicación 2. Esperar a que la pantalla de bienvenida termine de inicializar la aplicación. 3. Pronunciar el comando “buscar”. 4. Esperar a que la pantalla del buscador se muestre. 5. Pronunciar el comando “volver”. |
| Datos para el test | N/A |
| Resultado Esperado | La aplicación entiende el comando, y cierra la pantalla del “buscador”, volviendo a la pantalla principal. |
| Resultado Obtenido | La aplicación entiende el comando, y vuelve a la pantalla inicial. |
| Resultado del Test (Pasó/Falló) | Pasó |

| | |
|-----------|--|
| Nombre | TC07 - Búsqueda |
| Escenario | Luego de mostrarse la pantalla del buscador, el usuario quiere volver a la pantalla principal. |

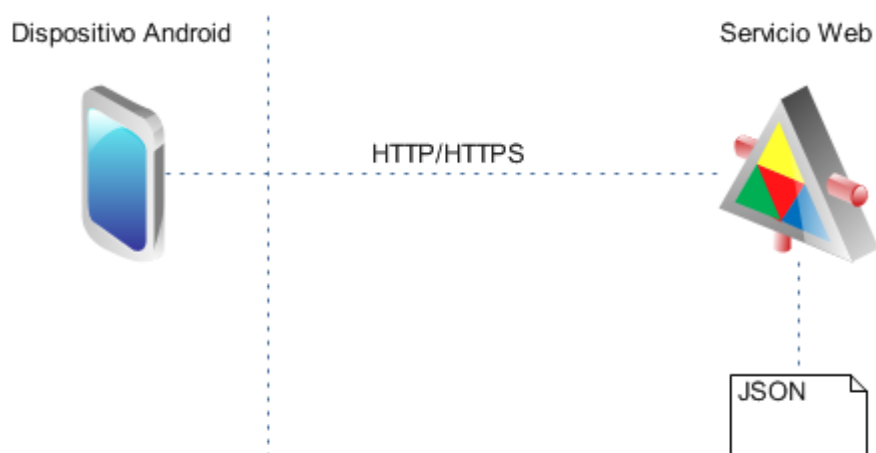
| | |
|---------------------------------|---|
| Pasos para el test | <ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Esperar a que la pantalla de bienvenida termine de inicializar la aplicación. 3. Pronunciar el comando “buscar”. 4. Esperar a que la pantalla del buscador se muestre. 5. Indicar una palabra clave del proceso que se desea buscar. |
| Datos para el test | <p>Los procesos disponibles se encuentran guardados en una aplicación en la nube, y son consultados a través de un request HTTP. Pueden agregarse y eliminarse los procesos guardados en cualquier momento.</p> <p>La palabra clave no puede ser ninguna de las que ya están asignadas a comandos concretos descritos previamente (“comandos”, “ayuda”, “salir”, “volver”).</p> |
| Resultado Esperado | <p>La aplicación recibe la palabra clave, y ejecuta el request. Recupera los resultados, y dependiendo de la cantidad, reacciona de diferentes formas:</p> <ul style="list-style-type: none"> • De no encontrar resultados, lo informa. • De encontrar resultados, los informa uno por uno. <p>Independientemente del resultado, la aplicación se queda esperando una nueva búsqueda.</p> |
| Resultado Obtenido | La aplicación reacciona correctamente, informando los resultados encontrados, e informando cuando no se encuentran coincidencias. |
| Resultado del Test (Pasó/Falló) | Pasó |

Arquitectura de la Solución

Diseño Básico

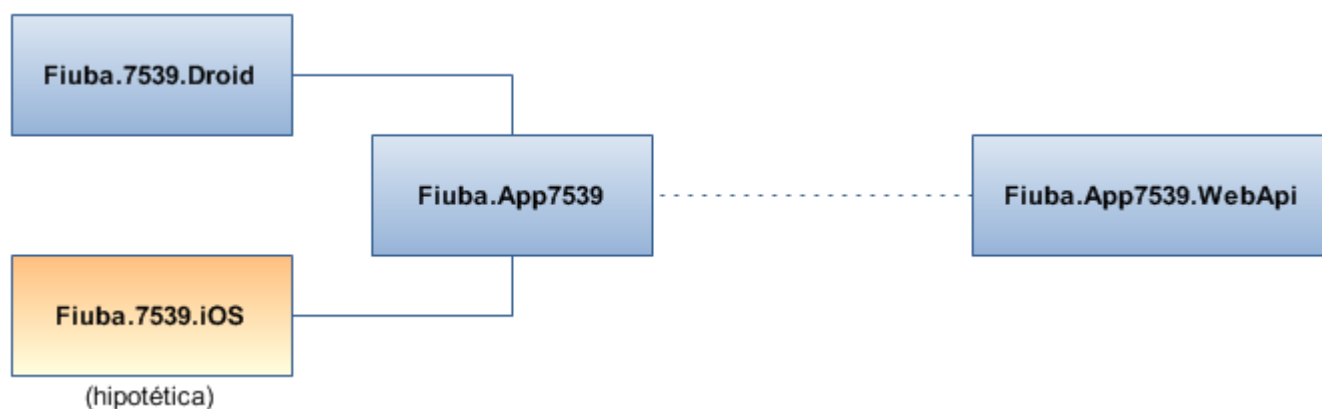
A continuación se muestra un esquema básico de la solución. La Aplicación para Android se instala en el teléfono y se comunica vía HTTP/HTTPS (depende de la implementación), con un servicio web que expondrá la información de los procesos que se pueden seguir dentro de la aplicación.

La información de los procesos se encuentra almacenados en un archivo de texto plano en formato JSON (<https://www.json.org/json-en.html>), incluyendo imágenes, codificadas en Base64. Una implementación distinta, podría contar con una base de datos, o alguna otra fuente de información.



Paquetes

La aplicación Android tiene dos componentes principales. Por un lado, la parte “nativa” de Android. Y por otro lado, una librería genérica que contiene los DTO que maneja la aplicación, así como algunos helpers que podrían ser reutilizados si se decidiera construir una versión para iOS, por ejemplo.



El tercer componente, es el servicio web con el que se comunica la aplicación para descargar la información sobre los procesos. Debe estar hosteado en un servidor donde pueda ser accedido mediante requests HTTP.

Temporizadores (desarrollo futuro)

Cada paso de los procesos debería contemplar la posibilidad de definir un tiempo determinado para su ejecución. Por ejemplo, en este proceso, puede verse que uno de los pasos tiene indicado un atributo con una cantidad de “segundos” que definirá esa temporización.

| | |
|------------------------|---|
| Id: | "3e327802-923d-eb11-ae60-e8de27783fba" |
| Name: | "Preparación de Fideos" |
| Items: | |
| ▶ 0: | {-} |
| ▶ 1: | {-} |
| ▼ 2: | |
| Id: | "b9454d53-566f-eb11-85e6-00163cc7547a" |
| Order: | 10 |
| Name: | "Calentar el agua" |
| Description: | "Poner la olla en la hornalla y encenderla." |
| Seconds: | 0 |
| ▶ Image: | "iVBORw0KGgoAAAANSUheUGAA_UBv5gAAAA8JRUS5ErkJggg==" |
| ▼ 3: | |
| Id: | "ff7d3176-566f-eb11-85e6-00163cc7547a" |
| Order: | 50 |
| Name: | "Esperar a que el agua hierva." |
| Description: | null |
| Seconds: | 600 |
| ▶ Image: | "iVBORw0KGgoAAAANSUheUGAA_buxt1MAAAAASUVORK5CYII=" |
| ▶ 4: | {-} |
| ▶ 5: | {-} |
| ▶ 6: | {-} |
| ▶ 7: | {-} |
| ▶ 8: | {-} |
| ▶ Requirements: | {-} |

Los pasos que no requieran un temporizador no tendrán este atributo al consultar a la aplicación, o bien deberían tener el valor 0 por default.

Esta implementación si bien exhibe en el servicio esa propiedad, es ignorada por la aplicación.

Video Demostrativo

Se puede comprobar el correcto funcionamiento de la aplicación

<https://www.youtube.com/watch?v=Eca3tCusHrY>

Repositorio

Todo el código es accesible en la siguiente repositorio:

<https://github.com/adri-kriptok/tpai7539/>

Hosting

El servicio web se encuentra actualmente hosteado en:

<http://fiuba7539-91186.somee.com/test>