Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc448254544)

[1.1 Autores 2](#_Toc448254545)

[1.2 Planificación 2](#_Toc448254546)

[1.3 Entrega 2](#_Toc448254547)

[2. Requisitos del prototipo a implementar 3](#_Toc448254548)

[2.1 Requisitos funcionales 3](#_Toc448254549)

[2.2 Otros requisitos 3](#_Toc448254550)

[3. Criterios de comparación en la implementación 4](#_Toc448254551)

[3.1 Criterio 1: Nombre del criterio 4](#_Toc448254552)

[3.2 Criterio 2: Nombre del criterio 4](#_Toc448254553)

[3.N Criterio N: Nombre del criterio 4](#_Toc448254554)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A 5](#_Toc448254555)

[4.1 Documentación de diseño 5](#_Toc448254556)

[4.2 Documentación de construcción 5](#_Toc448254557)

[4.3 Documentación de pruebas 5](#_Toc448254558)

[4.4 Documentación de instalación 5](#_Toc448254559)

[4.5 Manual de usuario 5](#_Toc448254560)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B 6](#_Toc448254561)

[5.1 Documentación de diseño 6](#_Toc448254562)

[5.2 Documentación de construcción 6](#_Toc448254563)

[5.3 Documentación de pruebas 6](#_Toc448254564)

[5.4 Documentación de instalación 6](#_Toc448254565)

[5.5 Manual de usuario 6](#_Toc448254566)

[6. Comparación de las dos implementaciones 7](#_Toc448254567)

[6.1 Evaluación de los criterios en la implementación usando la tecnología A 7](#_Toc448254568)

[6.2 Evaluación de los criterios en la implementación usando la tecnología B 7](#_Toc448254569)

[7. Comparación de la implementación de las tecnologías 8](#_Toc448254570)

[8. Conclusiones 10](#_Toc448254571)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

Grupo M1

Amir Sholkami Berube - Amir95sb(coordinador)

Javier Ballesteros García - JavierBallesteros5

Ismael Sainz Maza Jiménez - Ismaeel19

Stephany Susan Vela Vellachich - stephanyvela

## 1.2 Planificación



https://app.teamweek.com/#pg/1g82G63vww4N07cj0sN0WsaZLnDfTiuz

## 1.3 Entrega

https://github.com/Amir95sb/TG3

# 2. Requisitos del prototipo a implementar

## 2.1 Requisitos funcionales

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF01 | Testing de sumas y restas |
| RF02 | Testing Multiplicaciones y divisiones |
| RF03 | Guardar Procedimientos Test |

## 2.2 Otros requisitos

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| R01 | Testing de interfaz de usuario |
| R02 | Velocidad de pruebas >1 minuto |
|  |  |

# 3. Criterios de comparación en la implementación

***3.1 Interfaz de Usuario***

Nombre del criterio: Interfaz de Usuario.

Descripción: En este criterio comparamos la facilidad de la interfaz de usuario con respecto a otras aplicaciones. Más tarde veremos esa comparación de nuestros dos software frente a frente

Tipo de valor: Texto.

***3.2 Tiempo de Aprendizaje***

Nombre del criterio: Tiempo de Aprendizaje.

Descripción: En este criterio valoramos el tiempo que hemos tardado desde que nosotros arrancamos desde el momento que arrancamos nuestro software y podemos empezar a realizar ese proyecto. También introducimos el tiempo que le hemos dedicado a la adquisición de conocimientos de aprendizaje del software.

Tipo de valor: Numérico (horas).

***3.3 Tiempo de Configuración***

Nombre del criterio: Tiempo de Configuración.

Descripción: Incluimos el tiempo que hemos empleado en la configuración de nuestro software para utilizarlo. Aquí debemos incluir la instalación.

Tipo de valor: Numérico (horas).

***3.4 SS.OO Útiles***

Nombre del criterio: SS.OO Útiles.

Descripción: Diremos los SS.OO que hemos usado en la instalación y realización del proyecto. En este punto introducimos también la dificultad que hemos tenido.

Tipo de valor: Texto.

***3.5 Horas Empleadas en el Desarrollo***

Nombre del criterio: Horas Empleadas en el Desarrollo.

Descripción: Decimos el número de horas que se hemos tardado desde que empezamos en su momento el desarrollo hasta tener un proyecto viable al menos mínimamente.

Tipo de valor: Numérico (horas).

***3.6 Herramientas Utilizadas en el Desarrollo***

Nombre del criterio: Herramientas Utilizadas en el Desarrollo.

Descripción: Vemos las herramientas que hemos usado pues nos ayudaban a realizar ese mínimo de nuestro proyecto.

Tipo de valor: Texto.

***3.7 Costes del software***

Nombre del criterio: Costes del software

Descripción: Decimos cuanto a costado el software con todas las herramientas que también usamos para hacerlo funcionar.

Tipo de valor: Valor económico.

***3.8 Requisitos en el ordenador***

Nombre del criterio: Requisitos en el ordenador

Descripción: Decimos cuales son los requisitos para nuestro ordenador o dispositivo para que nuestro software pueda funcionar a pleno rendimiento.

Tipo de valor: Texto.

# 4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología A, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 4.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

## 4.2 Documentación de construcción

Hay que incluir una descripción de la construcción del prototipo, incluyendo algún extracto de código fuente. No es necesario todo el código. Sólo algún extracto para ver cómo se ha comentado.

## 4.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los caso de prueba.

## 4.4 Documentación de instalación

Para que Appium funcione correctamente en Windows debemos tener instalado NodeJs, Java Development Kit (JDK), Paquetes de Android SDK y ADB, además de los drivers de los dispositivos que usaremos, ya que Appium los necesita para conectarse y probar la aplicación. En caso de no poseer un dispositivo se requiere un emulador.

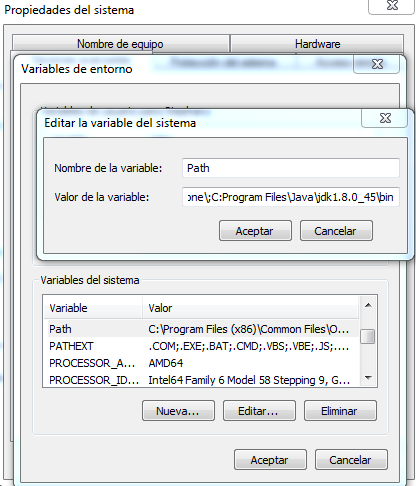
Para saber la versión de node.js (se recomienda tener al menos la v4.4.7):

node -v

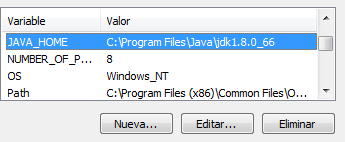
Para la versión de java (se recomienda tener a partir de la v1.8.0):

java –version

En caso de que desde la cmd no se permita ver la versión de java, deberemos indicar el camino. Para ello iremos a Panel de Control > Sistema y Seguridad > Sistema y entraremos a la pestaña de Configuración avanzada del sistema.

Desde opciones avanzadas daremos click a Variables de entorno y añadiremos a Path el camino al bin de nuestro software Java, pero sin quitar los demás ya que son necesarios.

Además añadiremos la variable JAVA\_HOME:



## 4.5 Manual de usuario

Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología B, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 5.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

## 5.2 Documentación de construcción

Hay que incluir una descripción de la construcción del prototipo, incluyendo algún extracto de código fuente. No es necesario todo el código. Sólo algún extracto para ver cómo se ha comentado.

## 5.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los caso de prueba.

## 5.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

## 5.5 Manual de usuario

Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 6. Comparación de las dos implementaciones

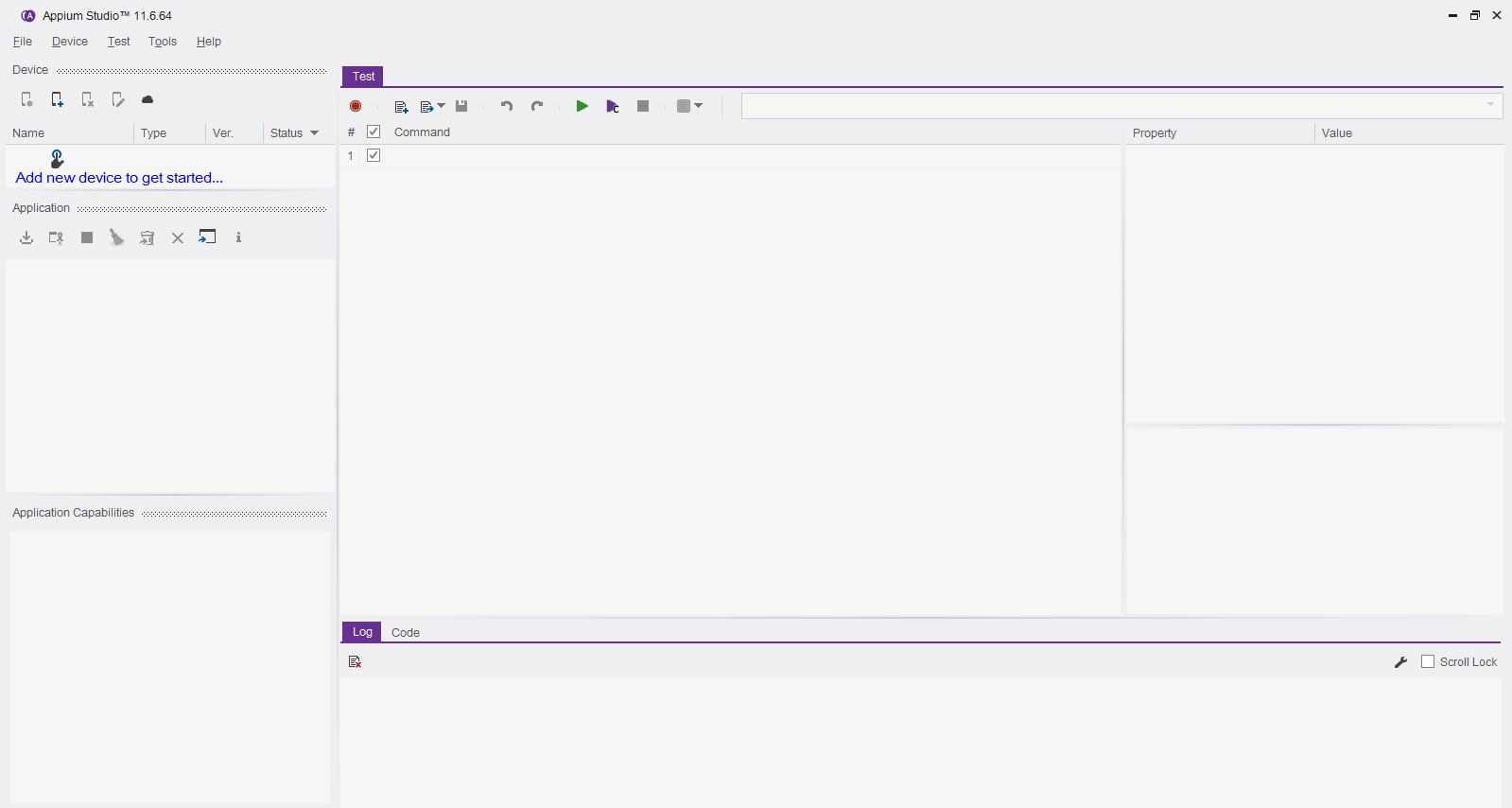
Se trata de dar valores a los criterios de comparación definidos en el apartado 3 sobre la implementación de cada uno de los prototipos.

## 6.1 Evaluación de los criterios en la implementación usando la tecnología A

Debe incluir al menos una tabla con la siguiente estructura.

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Interfaz de Usuario | Al abrir la aplicación se nos presenta una interfaz sencilla y funcional. Podemos añadir nuevos dispositivos, abrir proyectos, etc. La interfaz es clara.\* |
| Tiempo de aprendizaje | 3 horas (aprox) |
| Tiempo de configuración | 5 horas (aprox)\*\* |
| Sistema Operativos útiles | Se ha usado desde Windows para probar una apk de Android. |
| Horas empleadas en el desarrollo | Appium Studio, Android SDK, NodeJs, drivers para Xperia Z1\*\* |
| Herramientas utilizadas en el desarrollo | 5 horas |
| Costes de software | 0 |
| Requisitos en el ordenador | Se requieren diversos software complementarios y drivers, además de un dispositivo o emulador con el sistema operativo deseado.\*\* |

\*Pantalla de la interfaz de usuario:



\*\* Al instalar Appium y querer hacer una prueba sobre Android debemos instalar varios complementos, como el Java Development Kit (JDK), Android SDK/ADB y los paquetes deseados, NodeJs, Microsoft .net Framework antes de ponernos a trabajar. Estas instalaciones pueden llevar su tiempo, ya que son bastantes. Por ello, la habilidad a la hora de configurarlo es elevada, ya que además se requiere la preparación de los elementos como se muestra en las próximas capturas de pantalla.

Además, para poder usar el software se nos pide que instalemos unos drivers, es decir, necesitamos tener un dispositivo o un emulador para poder probar el código.

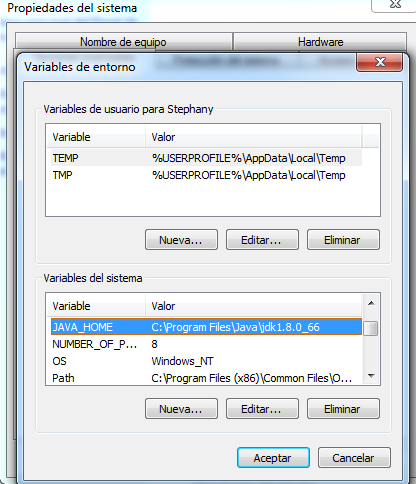
Si no se tiene alguno de los complementos o drivers, es seguro o muy probable que no se pueda realizar la prueba.

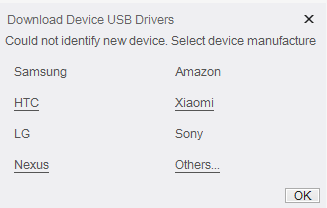
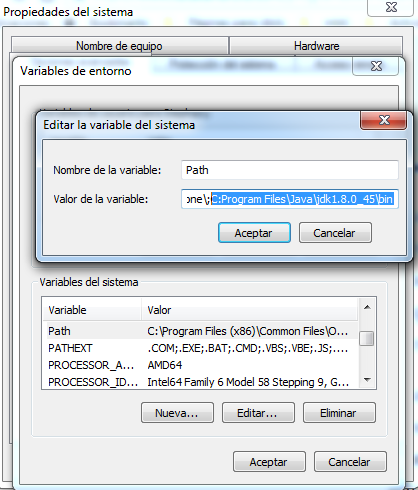
Se han tenido diversos problemas con la instalación de Appium por lo que su configuración ha durado más de lo esperado.

En cuanto a los requisitos de hardware:

- Requisitos mínimos:

* Procesador Intel®  i5 or i7
* 1 GB de espacio en el disco duro.
* 8 GB RAM
* Un Puerto USB 2.0 y un cable USB





## 6.2Evaluación de los criterios en la implementación usando la tecnología B

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Interfaz de Usuario | Es muy completa ya que al estar Espresso integrado a Android Studio, las opciones son infinitas. Puedes diseñar la aplicación directamente, y ejecutar las pruebas sin cambiar de aplicación. |
| Tiempo de aprendizaje | 10 horas (aprox) |
| Tiempo de configuración | 2 horas (aprox)\*\* |
| Sistema Operativos útiles | Se ha usado desde Windows para probar una apk de Android. |
| Horas empleadas en el desarrollo | 20 horas |
| Herramientas utilizadas en el desarrollo | Android Studio, Java JDK's y Android SDK's |
| Costes de software | 0 |
| Requisitos en el ordenador | Muchos. Requiere bastante capacidad de procesamiento.\* |

Requisitos en el ordenador.

- Requisitos mínimos:

* SSD
* 8 GB RAM
* Core i5

- Requisitos recomendados:

* SSD
* 16 GB RAM
* Core i7

Apuntar que aunque estos son los requisitos recomendados, hemos probado la herramienta en un portátil con los requisitos mínimos y no lo hemos conseguido hacer funcionar. Al cambiar a un ordenador más potente de sobremesa hemos conseguido hacerlo funcionar pero con problemas.

# 

# 7. Comparación de la implementación de las tecnologías

| **CRITERIOS** | **TECNOLOGÍA A** | **TECNOLOGÍA B** | **COMENTARIOS** |
| --- | --- | --- | --- |
| Interfaz de Usuario | Al abrir la aplicación se nos presenta una interfaz sencilla y funcional. Podemos añadir nuevos dispositivos, abrir proyectos, etc. La interfaz es clara.\* | Es muy completa ya que al estar Espresso integrado a Android Studio, las opciones son infinitas. Puedes diseñar la aplicación directamente, y ejecutar las pruebas sin cambiar de aplicación. | Hablando de la interfaz de usuario, la interfaz de Appium está muy cuidada con pocas opciones y claras mientras que Espresso tiene más opciones con una interfaz algo más tosca. |
| Tiempo de aprendizaje | 8 horas (aprox) | 10 horas (aprox) | Estas horas son aproximadamente las necesarias para poder comenzar a utilizar estas aplicaciones. Si quisiéramos utilizarlas con un mayor potencial necesitaríamos muchas más horas.  También destacar que Appium solo se dedica al testing de aplicaciones mientras que Espresso hace un desarrollo al detalle de las aplicaciones por lo que es más complicado. También nos da más facilidades según vamos aprendiendo a hacer los test. |
| Tiempo de configuración | 5 horas (aprox)\*\* | 2 horas (aprox)\*\* | Espresso es una aplicación integrada a Android Studio, y las herramientas SDK se instalan junto con Android Studio, mientras que en Appium, se necesitan muchas más herramientas y el proceso no es automatizado. |
| Sistema Operativos útiles | Se ha usado desde Windows para probar una apk de Android.  Soporta: Mac OS a partir de X 10.7, Windows 7/8/10 32 o 64 y Linux. | Se ha usado desde Windows para probar una apk de Android.  Soporta: Mac OS a partir de X 10.10 hasta 10.13, Windows 7/8/10 32 o 64 y Linux. | Las dos soportan los tres sistemas operativos más importantes por lo que no hay mucho que comentar en este apartado. |
| Horas empleadas en el desarrollo | 5 horas | 20 horas | Son las horas transcurridas durante varios días de visualizaciones de tutoriales por internet y manuales. Creíamos que era más fácil poner a funcionar estas tecnologías pero sin un conocimiento anterior son tecnologías que para alguien que no sepa nada antes se pueden hacer bastante complicadas.  Sobre todo hablamos de la tecnología Espresso la cual nos llevó 3-4 días saber como utilizarla. |
| Herramientas utilizadas en el desarrollo | Appium Studio, Android SDK, NodeJs, drivers para Xperia Z1\*\* | Android Studio, Java JDK's y Android SDK's | Son las herramientas necesarias para poder hacer funcionar estas tecnologías. |
| Costes de software | 0 | 0 | Todo el software utilizado es gratis. |
| Requisitos en el ordenador | Se requieren diversos software complementarios y drivers, además de un dispositivo o emulador con el sistema operativo deseado.\*\* | Muchos. Requiere bastante capacidad de procesamiento.\* | Hay una notable diferencia en cuanto a recursos hardware se requieren ya que Espresso pide unos requisitos bastante elevados y aun así no funciona correctamente. Con Appium ningún problema.  Esta diferencia se ve reflejada en la potencia de cada tecnología, siendo Espresso mucha más rápida realizando pruebas. |

# 8. Conclusiones

Por último, terminamos con la conclusión habiendo probado las dos tecnologías.

Es difícil decidirse por una tecnología sin saber las necesidades de cada uno ya que son herramientas que en principio realizan la misma función pero difieren mucho.

Si necesitamos una herramienta potente, con muchas posibilidades tanto de testeo como de desarrollo utilizaríamos Espresso ya que nos da estas opciones pero ya sabemos que la configuración de esta herramienta y su puesta en marcha es costosa. A la larga nos va a ser muy rentable esta herramienta pero necesitamos ordenadores potentes para poder hacerla funcionar, lo que nos permite realizar test a una increíble velocidad. Todo esto sabiendo que solo podemos utilizar esta tecnología con Android.

En cuanto a Appium, la escogeríamos si necesitáramos solo realizar pruebas y más sencillas, pero a la vez más versátiles por su capacidad de operar en distintos SO móviles y en muchos dispositivos a la vez. Además es más fácil de utilizar en un primer momento.

Por esto es que no podemos elegir una de las dos tecnologías sin saber para que las queremos con mas detalle.