Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc445388848)

[1.1 Autores 2](#_Toc445388849)

[1.2 Planificación 2](#_Toc445388850)

[1.3 Entrega 2](#_Toc445388851)

[2. Descripción de las tecnologías 3](#_Toc445388852)

[2.1 Descripción de la tecnología 1 3](#_Toc445388853)

[2.2 Descripción de la tecnología 2 3](#_Toc445388854)

[3. Criterios de comparación 4](#_Toc445388855)

[3.1 Categoría A: Nombre 4](#_Toc445388856)

[3.1.1 Criterio A.1: Nombre 4](#_Toc445388857)

[3.1.2 Criterio A.2: Nombre 5](#_Toc445388858)

[3.1.n Criterio A.n: Nombre 5](#_Toc445388859)

[3.2 Categoría B: Nombre 5](#_Toc445388860)

[3.2.1 Criterio B.1: Nombre 5](#_Toc445388861)

[3.2.2 Criterio B.2: Nombre 5](#_Toc445388862)

[3.2.n Criterio B.n: Nombre 5](#_Toc445388863)

[3.3 Categoría Z: Nombre 5](#_Toc445388864)

[3.3.1 Criterio Z.1: Nombre 5](#_Toc445388865)

[3.3.2 Criterio Z.2: Nombre 5](#_Toc445388866)

[3.3.n Criterio Z.n: Nombre 5](#_Toc445388867)

[4. Evaluación de los criterios por tecnología 6](#_Toc445388868)

[4.1 Evaluación de los criterios para la tecnología 1 6](#_Toc445388869)

[4.2 Evaluación de los criterios para la tecnología 2 6](#_Toc445388870)

[5. Comparación de las tecnologías 7](#_Toc445388871)

[6. Recomendaciones 9](#_Toc445388872)

[6.1 Situación 1 9](#_Toc445388873)

[6.1.1 Descripción de la situación 9](#_Toc445388874)

[6.1.2 Recomendación de tecnología a utilizar 9](#_Toc445388875)

[6.2 Situación 2 9](#_Toc445388876)

[6.2.1 Descripción de la situación 9](#_Toc445388877)

[6.2.2 Recomendación de tecnología a utilizar 9](#_Toc445388878)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

En este apartado se debe indicar el número de grupo y los nombres de los autores, poniendo en primer lugar al coordinador del grupo.

## 1.2 Planificación

En este apartado se debe incluir un enlace (URL) compartido a la planificación del trabajo utilizando una herramienta online de diagramación Gantt (por ejemplo, GanttPro, versión gratuita).

Hay que tener en cuenta que cada participante del grupo debe tener asignadas tareas que sumen al menos 15 horas. El peso de este trabajo en la calificación total de la asignatura es de un 10%, por tanto requiere de una dedicación de 15 horas del total de 150 horas de la asignatura.

## 1.3 Entrega

En este apartado debe incluirse un enlace (URL) a un repositorio en GitHubo en BitBucket creado para el trabajo.

En dicho repositorio debe encontrarse, al menos los siguientes archivos en la rama máster:

* Trabajo terminado: del trabajo terminado con el nombre TG2\_final.docx
* Presentación del trabajo: TG2\_final.pptx

Dichos archivos serán los que se tendrán en cuenta para la calificación del trabajo.

# 2. Descripción de las tecnologías

## 2.1 Descripción de Appium.

### Appium:

[](http://appium.io/)

Appium es una herramienta de código abierto para automatizar aplicaciones nativas móviles en plataformas de escritorio con iOS, Android y Windows. Las aplicaciones nativas son aquellas escritas usando iOS, Android o SDK de Windows.

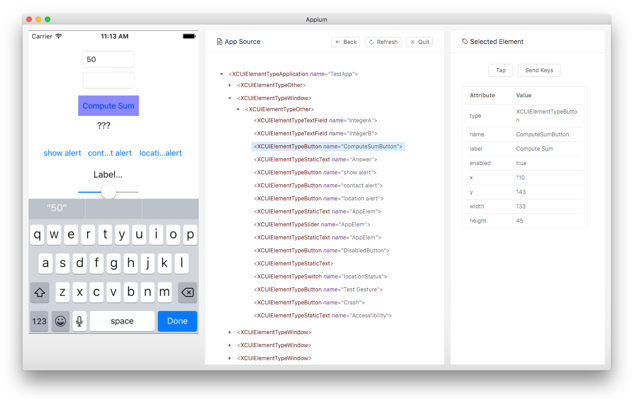
Es importante destacar que Appium es "multiplataforma": le permite escribir pruebas en múltiples plataformas (iOS, Android, Windows), utilizando la misma API. Esto permite la reutilización de códigos entre las salas de pruebas de iOS, Android y Windows.

También cabe resaltar que permite escribir los scripts en casi cualquier lenguaje de programación, por ejemplo Java, Objective-C, , JavaScript, PHP, Ruby, Python or C#, etc)

Filosofía de Appium:

Appium fue diseñado para satisfacer las necesidades de automatización móvil de acuerdo con una filosofía descrita por los siguientes cuatro principios:

* No debería tener que volver a compilar su aplicación o modificarla de ninguna manera para automatizarla.
* No debe estar encerrado en un idioma o marco específico para escribir y ejecutar sus pruebas.
* Un marco de automatización móvil no debe reinventar la rueda.
* Un marco de automatización móvil debe ser de código abierto, tanto en espíritu como en práctica.



2.2 Descripción de Espresso.

### Espresso:

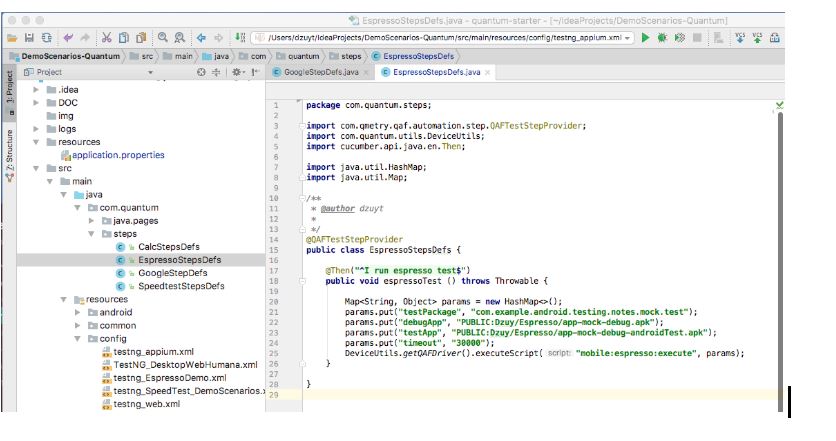
[](https://google.github.io/android-testing-support-library/docs/espresso/)

Espresso es una herramienta para empresas. Como herramienta de código abierto es muy fácil de usar y puede extenderse dentro del entorno de trabajo con facilidad.

Su principal ventaja es que nos permite la automatización automática de aplicaciones sin escribir ni una sola línea de código. Solo puede trabajar en el OS Android pero lo hace con mucha más velocidad y eficiencia que otras tecnologías.

Espresso permite realizar pruebas tanto en dispositivos físicos como virtuales ([emuladores](https://es.wikipedia.org/wiki/Emuladores)).La principal desventaja de ejecutar las pruebas en dispositivos físicos (reales) se debe a su alto costo y tiempo que requiere por la gran variedad de dispositivos existentes en la actualidad. Una posible solución es ejecutar estas pruebas en un emulador (ambiente controlado que permite soportar múltiples versiones de [SO](https://es.wikipedia.org/wiki/SO), tamaños de pantalla y limitantes de memoria).

A modo de ejemplo, si comparamos la velocidad de ejecución de las pruebas en un Nexus 4 contra un emulador, este último requiere un 65% menos de tiempo para llevarla a cabo. De acuerdo al equipo de trabajo detrás de Espresso, su solución logró detectar el 99% de los bugs en las aplicaciones Android, dejando unos pocos para las pruebas tradicionales (reduciendo el esfuerzo requerido para las mismas).



# 3. Criterios de comparación

En los sub-apartados de este apartado se deben indicar cada uno de los criterios (también llamados factores, propiedades, características, indicadores, etc.), que se usarán en la comparación.

Los criterios deben organizarse en categorías (al menos 3 categorías). El número de criterios totales no puede ser inferior a 20.

Las categorías dependerán del tipo de tecnología, pueden ser categorías como “General”, “Utilidades”, “Rendimiento”, etc.

Los criterios a definir en cada categoría también dependerán del tipo de tecnologías a comparar. En el siguiente apartado hay algunos ejemplos.

## 3.1 Categoría A: Nombre

### 3.1.1 Criterio A.1: Nombre

Por cada criterio hay que indicar el nombre, una breve descripción, y el tipo de valor a asignar al criterio. El número de criterios no puede ser inferior a 20.

Por ejemplo, si se comparan dos herramientas CASE, un criterio podría ser el autor, entonces en este apartado se podría incluir este texto:

*3.1.1Criterio A.1: Autor de la herramienta*

*Nombre del criterio: Autor.*

*Descripción: Nombre de la persona, institución o empresa que ha creado la herramienta.*

*Tipo de valor: Texto libre.*

Otro ejemplo de criterio con valor Si/No:

*3.1.1CriterioA.1: Generación de código Java*

*Nombre del criterio: Generación código Java.*

*Descripción: Indica si la herramienta CASE incluye funcionalidad para generar archivos fuente .java a partir de diagramas de clase.*

*Tipo de valor: Booleano (Si/No).*

Otro ejemplo de criterio con valor numérico:

*3.1.1CriterioA.1: Número de proyectos simultáneos*

*Nombre del criterio: Proyectos simultáneos.*

*Descripción: Indica el número máximo de proyectos de modelado que deja tener abiertos la herramienta de forma simultánea.*

*Tipo de valor: Numérico.*

### 3.1.2 Criterio A.2: Nombre

### 3.1.n Criterio A.n: Nombre

## 3.2 Categoría B: Nombre

### 3.2.1 Criterio B.1: Nombre

### 3.2.2 Criterio B.2: Nombre

### 3.2.n Criterio B.n: Nombre

## 3.3 Categoría Z: Nombre

### 3.3.1 Criterio Z.1: Nombre

### 3.3.2 Criterio Z.2: Nombre

### 3.3.n Criterio Z.n: Nombre

# 4. Evaluación de los criterios por tecnología

## 4.1 Evaluación de los criterios para la tecnología 1

Debe incluir al menos una tabla con la siguiente estructura.

|  |  |
| --- | --- |
| CRITERIOS | EVALUACIÓN |
| Criterio A.1: Nombre |  |
| Criterio A.2: Nombre |  |
| Criterio A.n: Nombre |  |
| … |  |
| Criterio B.1: Nombre |  |
| Criterio B.2: Nombre |  |
| … |  |

Y algunos comentarios aclaratorios sobre aquellos criterios cuyo valor indicado en la tabla no sea suficiente para entenderlo.

La tabla anterior es obligatoria y deben completarla los autores del trabajo, aunque se pueden incluir otros gráficos o tablas complementarias copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada uno se indique la fuente (al menos la URL).

## 4.2Evaluación de los criterios para la tecnología 2

# 5. Comparación de las tecnologías

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

Esta tabla anterior es obligatoria y deben completarla los autores del trabajo, aunque se pueden incluir otros gráficos o tablas complementarias copiadas y pegadas desde diversas fuentes de información, siempre que debajo de cada uno se indique la fuente (al menos la URL).

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERIOS | TECNOLOGÍA A | TECNOLOGÍA B | COMENTARIOS |
| A.1 |  |  |  |
| A.2 |  |  |  |
| … |  |  |  |
| B.1 |  |  |  |
| B.2 |  |  |  |
| … |  |  |  |

# 6. Recomendaciones

Deben platearse posibles situaciones de uso, y recomendar justificadamente una u otra tecnología en función de la situación. Al menos 2 situaciones diferentes.

## 6.1 Situación 1

### 6.1.1 Descripción de la situación

Una posible situación podría ser el de una empresa que realiza automatización de pruebas móviles para todo tipo de dispositivos y apps. Ya que estos dispositivos pueden tener distintos OS y estar escritos en diferentes lenguajes, la empresa busca una herramienta que le permita esta versatilidad a la hora de trabajar.

### 6.1.2 Recomendación de tecnología a utilizar

Appium permite realizar pruebas de aplicaciones en iOS, Android y Windows además de poder programar en un gran número de lenguajes, los mas utilizados. Esto permitiría a la empresa automatizar pruebas en la gran mayoría de dispositivos por la parte del OS y también por los distintos lenguajes utilizados en esas aplicaciones al soportar varios lenguajes.

En el caso de Espresso estas dos ventajas no existen ya que solo permite aplicaciones para OS Android y no permite tantos lenguajes de programación

La tecnología usada en este caso sería **Appium.**

|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Ventajas Appium | Ventajas Espresso |
| Multiplataforma | Si | No |
| Variedad de lenguajes de programación | Si | No |

## 6.2 Situación 2

### 6.2.1 Descripción de la situación

Una empresa quiere empezar un proyecto de automatización de pruebas de aplicaciones móviles en sistemas operativos Android y para ello confía a un jefe de proyecto la búsqueda de una herramienta que se adapte a sus necesidades.

Necesita una herramienta fácil de aprender, que se adapte al entorno de trabajo fácilmente para poder empezar el proyecto lo antes posible y que además cumpla con todos los requisitos necesarios para probar las aplicaciones a fondo.

### 6.2.2 Recomendación de tecnología a utilizar

La tecnología utilizada en este caso seria **Espresso.**

Appium puede probar de forma correcta las aplicaciones Android pero es más complicada de utilizar y no se adapta de forma tan fácil al entorno de trabajo.

Espresso en el caso contrario, además de ser la herramienta más rápida y eficiente en cuanto a las pruebas en OS Android, aunque solo pueda probar sobre este OS, se adapta rápidamente al área de trabajo con su API.

|  |  |  |
| --- | --- | --- |
| Criterios relevantes para la decisión | Ventajas Appium | Ventajas Espresso |
| Adaptación al entorno de trabajo | No | Si |
| Requisitos para las pruebas | Si | Si |
| Facilidad de uso | No | Si |

---------------------------

(Hay que cumplir la estructura básica indicada de secciones. Pero si se desea se pueden añadir otras secciones como anexos. Por ejemplo, alguna encuesta de opinión realizada sobre las tecnologías, etc.)