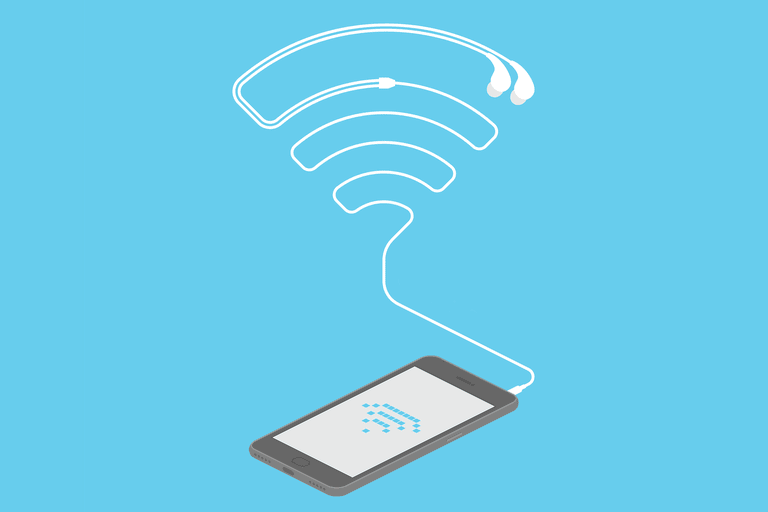
Trabajo de Fin de Grado

Aplicación móvil para la gestión de servidores Linux/UNIX



Álvaro Gómez Muñoz

Documento de Borrador V0.0.1

Este documento se presta a cambios en todo momento según se vea durante desarrollo, añadiendo nuevos apartados y al final será portado a Latex.

En la versión final de Latex se completaran los requisitos sobre la realización de memorias según la guía de propuestas de EINA.

Contenido

[Descripción del proyecto 3](#_Toc509243972)

[Análisis de Aplicaciones existentes 4](#_Toc509243973)

[Aplicaciones de Servidor 4](#_Toc509243974)

[Zabbix 4](#_Toc509243975)

[Nagios 5](#_Toc509243976)

[Aplicaciones Móviles 5](#_Toc509243977)

[Server & Website Monitor 6](#_Toc509243978)

[Cura SysAdmin 6](#_Toc509243979)

[Análisis y Desarrollo 7](#_Toc509243980)

[Objetivo de la aplicación 7](#_Toc509243981)

[Requisitos 7](#_Toc509243982)

[Apartado servidor 7](#_Toc509243983)

[Apartado cliente 7](#_Toc509243984)

[Diseño 7](#_Toc509243985)

[Desarrollo de API del servidor 7](#_Toc509243986)

[Usuarios de la aplicación 7](#_Toc509243987)

[Servicios 8](#_Toc509243988)

[Ficheros de logs 9](#_Toc509243989)

[Desarrollo de Aplicación Móvil 10](#_Toc509243990)

[Prototipo 10](#_Toc509243991)

[Bibliografica 15](#_Toc509243992)

# Descripción del proyecto

Se pretende diseñar una aplicación móvil que recoja en un mismo dispositivo, todos los posibles servidores Linux que un usuario pueda estar gestionando. De esta manera, se pretende que dicho usuario tenga facilidades a la hora de controlar de una manera fácil y cómoda varios aspectos de sus correspondientes servidores.

Dispondrá de distintas funcionalidades, entre ellas por ejemplo la inclusión de alertas sobre todos los aspectos que se estén monitorizando.

Otra de las funcionalidades que se incluirán, es la gestión de acciones rápida a través de botones personalizados, que pueden facilitar el trabajo al usuario.

Dicha aplicación irá acompañada de una API que se instalará en el servidor y que se encargara de recopilar toda la información que se monitorice y la almacenara a la espera de que el usuario desee consultarla, de tal manera que se minimizara lo máximo posible la transmisión de datos entre el servidor y la aplicación móvil.

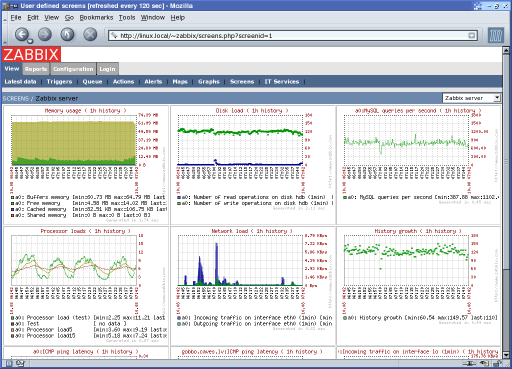
La API también será la encargada de gestionar las peticiones del usuario como por ejemplo el lanzamiento de comandos a través de los botones personalizados o también del envió de las alertas al móvil del usuario.

Aun existiendo aplicaciones en este segmento de mercado, casi todas se encuentran limitadas a la hora de monitorizar el sistema, esto es debido a que solo se dispone de la aplicación móvil, lo que provoca que el móvil no reciba mucha información o que este se encuentre con un uso excesivo uso de los datos de internet.

# Análisis de Aplicaciones existentes

## Aplicaciones de Servidor

### Zabbix

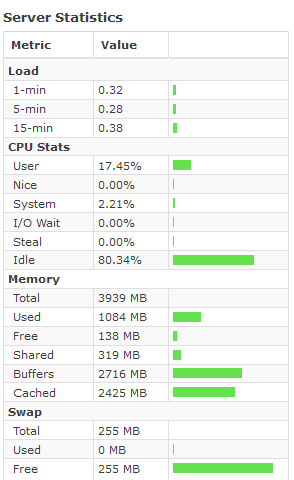


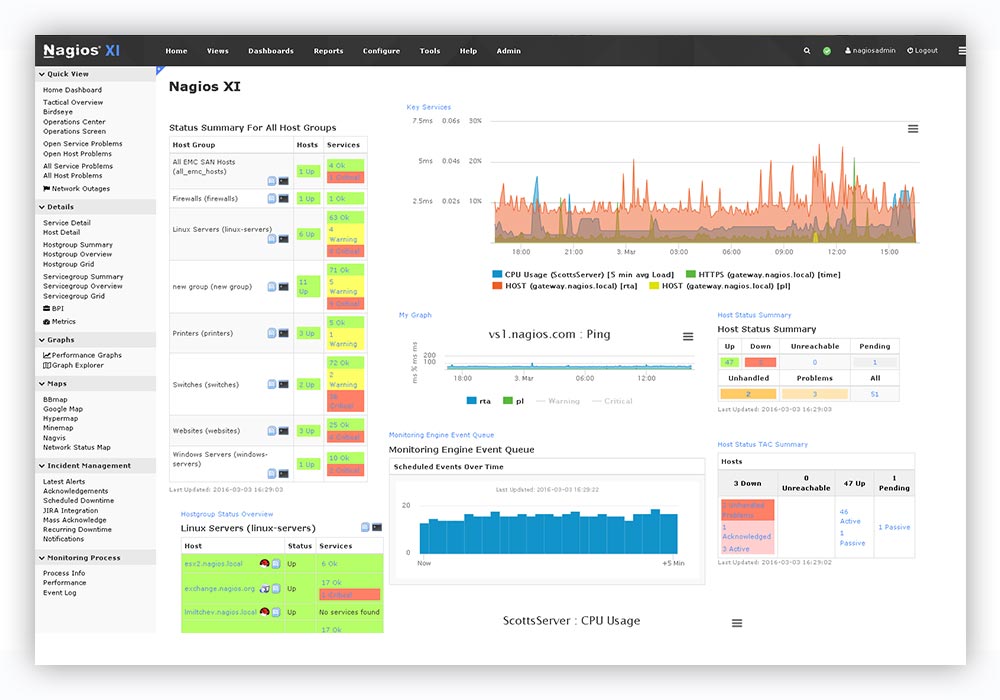
Es un Sistema de Monitorización de Redes creado por Alexei Vladishev. Está diseñado para monitorizar y registrar el estado de varios servicios de red, Servidores, y hardware de red. No dispone de aplicación móvil.

|  |
| --- |
| **Datos Monitorizados por zabix entre otros datos** |
| Uso de CPU |
| Uso de disco |
| Uso de Memoria |
| Uptime, Downtime |
| Dispone de trigger, eventos |
| Trafico de Red |

Info: <https://www.zabbix.com/documentation/3.4/manual/config/items/itemtypes/zabbix_agent>

### Nagios





Es un sistema de monitorización de redes ampliamente utilizado, de código abierto, que vigila los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado

|  |
| --- |
| **Datos Monitorizados por nagios entre otros datos** |
| Uso de CPU |
| Uso de disco |
| Uso de Memoria |
| Uptime, Downtime |
| Dispone de trigger, eventos |
| Trafico de Red |
| Usuarios Loggeados |

Info: <https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/monitoring-linux.html>

Nagio Demo: <http://nagiosxi.demos.nagios.com/>

## Aplicaciones Móviles

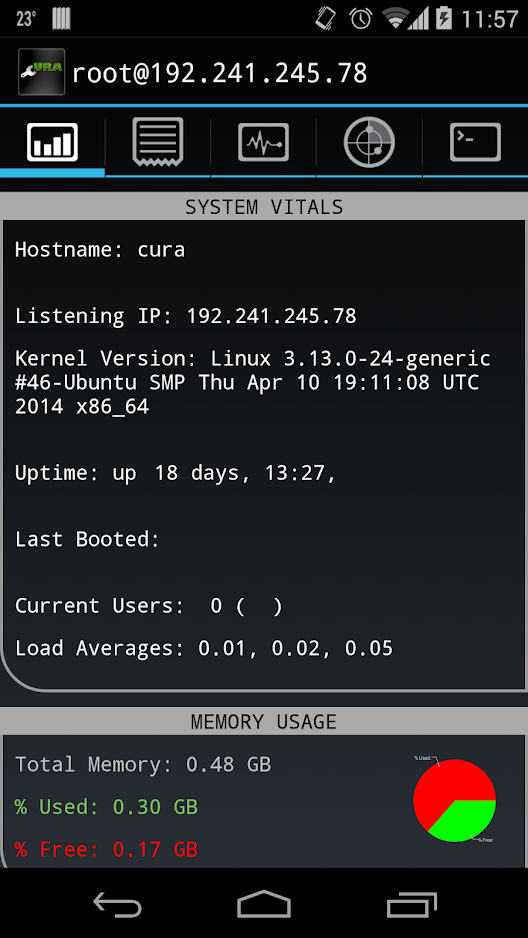
Ambas aplicaciones mencionadas en el apartado anterior disponen de aplicaciones móviles. Estas se encuentran limitadas, ya que han sido desarrolladas para un entorno más orientado a web. Estas limitaciones entran dentro de visualización y de funcionalidad en genera.

### Server & Website Monitor



Esta aplicaciones, está orientada únicamente saber si el servidor registrado, se encuentra en funcionamiento.

### Cura SysAdmin



Esta aplicación está orientada a ver un mínimo de información del sistema y lanzar comandos vía ssh. En la actualidad, esta aplicación ha dejado de actualizarse y está dejando de funcionar en ciertos dispositivos móviles, a la vez que le faltan algunas funcionalidades.

# Análisis y Desarrollo

## Objetivo de la aplicación

El objetivo de la aplicación es proporcionar al usuario una manera de ver el funcionamiento de sus servidores online y realizar acciones sobre estos de una manera fácil.

Los puntos fuertes de la aplicación que se va a desarrollar es entre otras el mínimo uso de los datos del dispositivo móvil y de disponer de varios perfiles de usuario, los cuales dispongan de permisos de control y monitorización o solo de monitorización.

También el desarrollar la aplicación con las ultimas tecnologías.

## Requisitos

[Se indicaran más adelante las versiones y más requisitos]

### Apartado servidor

* Nodejs
* Mysql

### Apartado cliente

* Android Versión X o superior

# Diseño

## Desarrollo de API del servidor

Viendo la información recopilada de las aplicaciones existentes, se va proporcionar una guía de cuáles serán los servicios que se van a proporcionar a través de servidor y que el dispositivo móvil utilizará.

Hay que tener en cuenta que la aplicación se encontrara limitada dentro de los permisos del usuario que ejecute esta API. Por lo tanto se recomienda que dicho usuario disponga como mínimo de permisos de lectura sobre los ficheros del apartado ficheros de logs que se comentaran más adelante y también sobre los comandos que se vayan a ejecutar a través de los botones personalizados.

### Usuarios de la aplicación

Existirán dos tipos de usuarios dentro de la aplicación.

|  |  |
| --- | --- |
| **Tipo** | **Rol** |
| Administradores | Tienen permisos para monitorizar el sistema y crear eventos o lanzar comandos. También disponen de permisos para crear usuarios dentro de la aplicación. |
| Monitores | Solo disponen de permisos de monitorización del servidor, sin ningún tipo de opción de lanzar comandos. |

Existirá un usuario de inicio, configurado previamente a inicial la aplicación.

### Servicios

Servicios de logueo

* Acceso a Administrador y Monitores

|  |  |
| --- | --- |
| **Servicios** | **Descripción** |
| /api/login | Permitirá al usuario loguarse dentro de la API y poder realizar operaciones dentro de esta. |

Servicios de configuración

* Acceso a Administrador

|  |  |
| --- | --- |
| **Servicios /api/config/** | **Descripción** |
| /api/config/configData | Permitirá configurar la aplicación con algunos parámetros de interés:   * Cada cuanto tiempo recogerá información la aplicación. |

Servicios de creación de usuarios

* Acceso a Administrador

|  |  |
| --- | --- |
| **Servicios /api/users/** | **Descripción** |
| /api/users/listUsers | Lista los usuarios de la aplicación y sus roles |
| /api/users/createUser | Creación de usuarios. |
| /api/users/deleteUser | Borrar un usuario creado por el admin |

Servicios de Monitorización

* Acceso a Administrador y Monitores

|  |  |
| --- | --- |
| **Servicios /api/look/** | **Descripción** |
| /api/look/cpu | Muestra el uso CPU Total y que procesos la cargan 5 maximo. |
| /api/look/disk | Muestra el uso del disco, espacio libre y espacio ocupado. |
| /api/look/memory | Muestra el uso de Memoria Total y por procesos 5 maximo. |
| /api/look/upDownTime | Muestra los datos de los últimos reinicios, apagados… del sistema. |
| /api/look/sockets | Muestra el estado de los sockets. Si alguno se encuentra activo, escuchando… |
| /api/look/netUsage | Muestra el uso de la red, cuantos datos se consumen y que procesos lo hacen. |
| /api/look/userLogin | Muestra información de los usuarios que se han conectado al sistema. |
| /api/look/processList | Muestra la lista de procesos funcionando en el sistema. |

Servicios de Comandos

* Acceso a Administrador

|  |  |
| --- | --- |
| **Servicios /api/event/** | **Descripción** |
| /api/run/listEvents | Muestra los eventos programados. |
| /api/run/createEvent | Creación de Eventos configurables para que se ejecuten en determinados momentos deseados. Dicho evento será un script a ejecutar.  Ejemplos:   * Uso de CPU superior a X % * Hora determinada |
| /api/run/deleteEvent | Borrado de un evento programado. |
| /api/run/listScripts | Lista los scripts dentro del sistema. |
| /api/run/createScript | Creación de script, se permite dos opciones:   * Subir un comando * Subir un fichero de script * Indicar dirección del script dentro del sistema. |
| /api/run/deleteScript | Borrar un script que se haya creado. |
| /api/run/launchScript | Ejecución de script, se permite dos opciones:   * Recibir un comando y ejecutarlo * Ejecutar un script ya subido |
| /api/run/killProcess | Permite cerrar un proceso ejecutándose en el sistema |

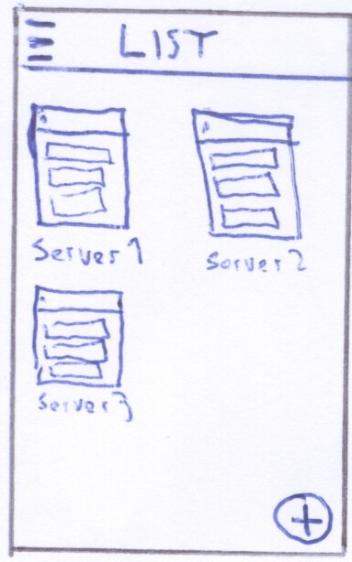
### Ficheros de logs

[Se completara durante el desarrollo cuando se vean cuales serán los ficheros requeridos]

## Desarrollo de Aplicación Móvil

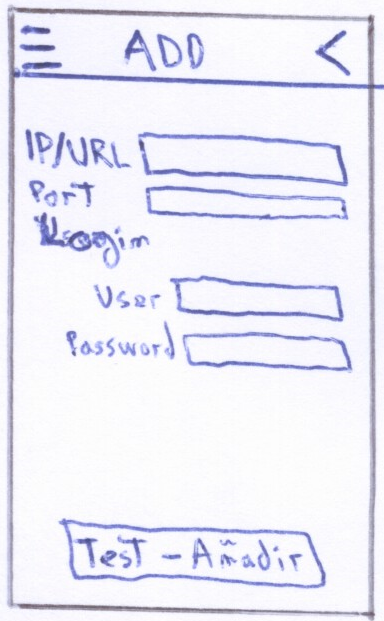
### Prototipo

El siguiente prototipo es una muestra de cómo será la aplicación, no es la solución final de este.

* Lista de servidores

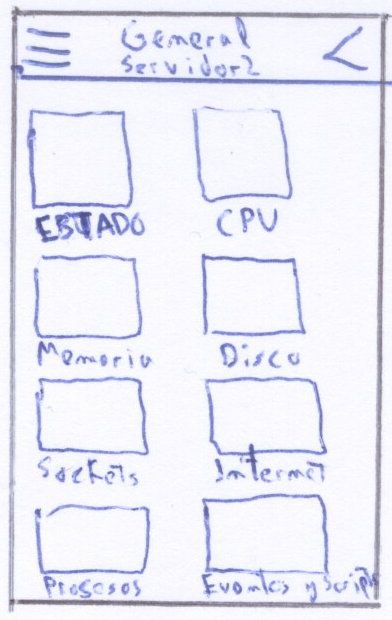
Desde esta pantalla se permitirá ver todos los servidores agregados a la aplicación, a la vez que acceder a cada uno y agregar nuevos.

* Añadir Nuevo Servidor

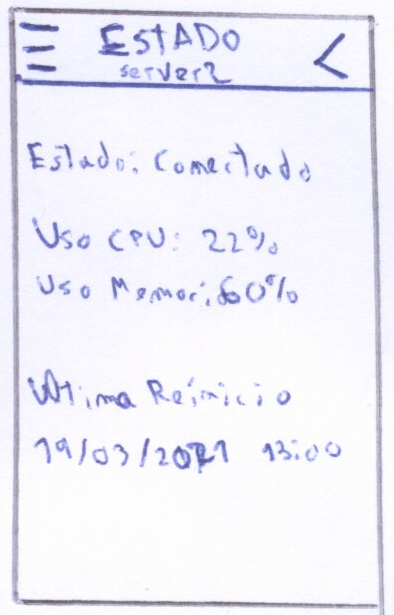


Desde esta pantalla se podrá agregar un nuevo servidor, con las correspondientes credenciales de este.

* Opciones dentro del servidor

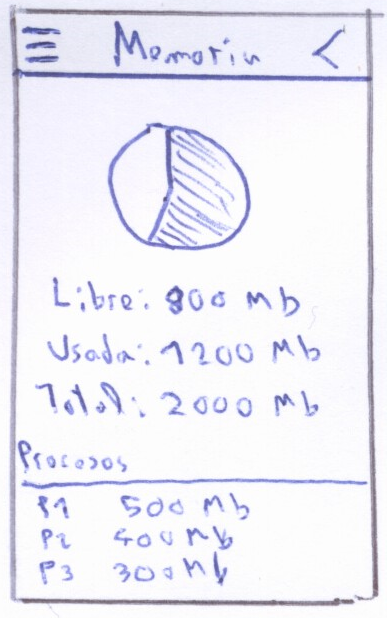


Este panel mostrara todas las opciones dentro del servidor, entre ellas las siguientes:

* Estado del servidor
* CPU
* Memoria
* Disco
* Sockets
* Consumo de Internet
* Procesos
* Eventos
* Scripts
* Usuarios
* Configuraciones
* Estado del servidor

Este panel mostrara información genérica del estado del servidor y alguna información extra que puede ser de utilidad.

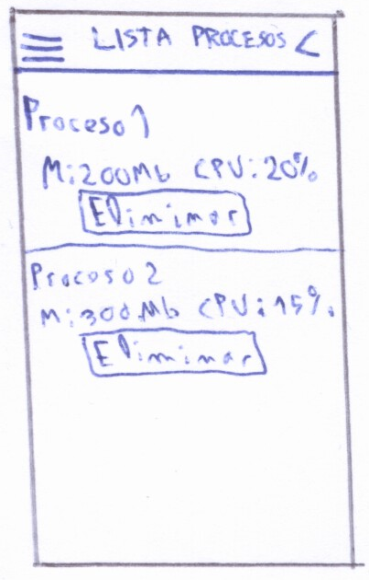
* Visualización de datos



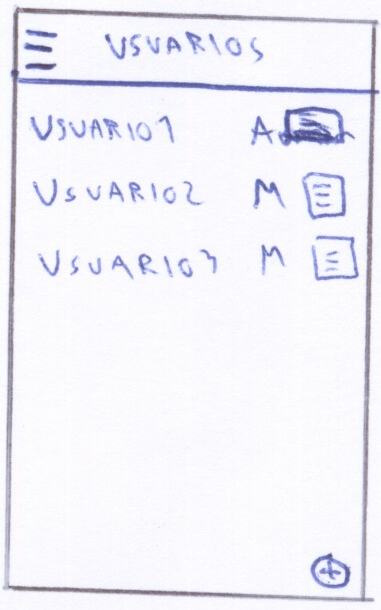
Desde esta pantalla se podra visualizar la memoria del sistema con los procesos que más consumen.

Esta pantalla será similar a las pantallas de visualización de CPU, Internet… pero con distintas graficas o distinta manera de visualizar los datos. De tal manera que cada uno disponga de la mejor manera de visualización.

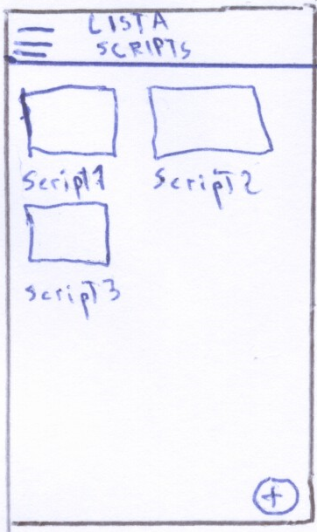
* Procesos del sistema

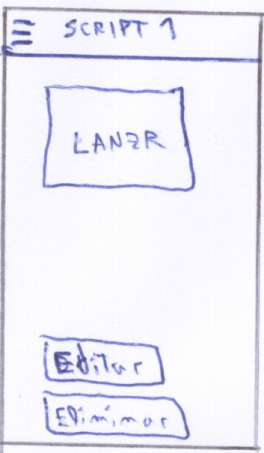


Esta pantalla mostrara los datos de cada proceso en el sistema y permitira Detener cada uno si se solicita.

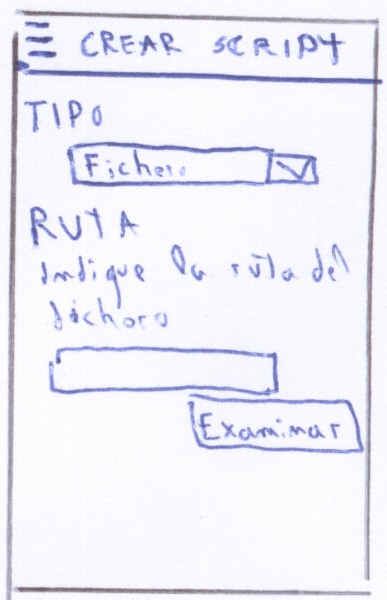
* Usuarios

Esta pantalla permitira visualizar cada usuario dentro de la API del servidor y su creación y modificación.

* Lista de Scripts

También definidos como los botones personalizados. Este panel permite ver todos los botones personalizados, permitiendo añadir nuevos botones.

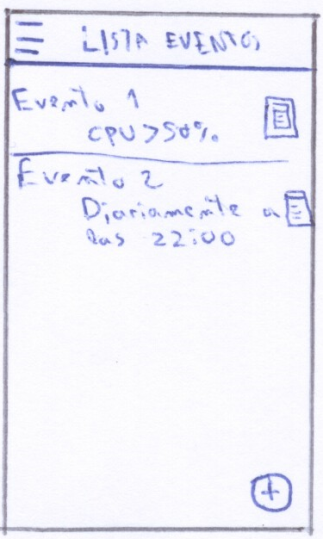
* Lanzamiento de Scripts

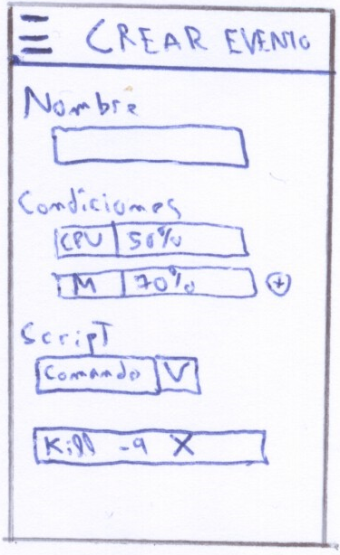
Panel que permite ejecutar el script seleccionado.

* Crear Script

Permite crear un nuevo script. Tiene 3 opciones posibles, subir un Fichero de script, ejecutar uno existente en el sistema o añadir una línea de comando.

* Lista de eventos

Panel que permite visualizar los eventos que se han preparado con sus respectivas condiciones de ejecución.

* Crear Evento

Permite crear nuevos eventos para que se ejecuten en caso de que se cumpla una condición. Las opciones de las condiciones, serán por ejemplo, CPU, Memoria…

# Desarollo

Este apartado explica como ha evolucionado el desarrollo de la aplicación y que tecnicas, librarias y frameworks se han usado para su desarrollo.

## Apartado Servidor

Durante el desarrollo del backend, se ha utilizado la tecnologia de NodeJs, puesto que es una tecnologia que actualmente se esta usando en profundida y tiene una comunidad muy amplia. Y con ellos, se han usado los sigueintes modulos:

### Framwork

Se ha utilizado el framework express, el cual se define a continuación alguno de sus beneficios:

* Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
* Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
* Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

Esto lo que permite es facilitar al desarrollador con alguno de los aspectos de la aplicación agilizando los procesos.

### Librerias

En este caso se va a destacar una de las librerias más importantes que se han utilizado dentro de la aplicación,

En este caso se ha utilizado “systeminformation”, una librería que se encarga de recopilar la información del sistema, incluyendo temperaturas, usuarios, discos…

Dicha librería esta desarrollada por **plusinnovations** con licencia MIT, dicha librería es muy completa y se ha integrado con la mayor parte de las funcionalidades de la aplciación.

A pesar de ser una librería muy completa, se encuentra en desarrollo, lo que a causado que varias de las funcionalidades deseadas no se encontren o en su defecto, no estuvieran terminadas. Por ello se ha apoyado al proyecto, lanzando una petición al usuario con modificaciones de código, las cuales el usuario ha aceptado y ha subido al repositorio.

Dichas modificaciones iban relacionadas con temas de devolver información relevante a usuarios con pocos permisos dentro del sistema. De tal manera que dichos usuarios no podrian actuar de monitores del sistema, al no poder recibir información de la librería.

# Apartado Cliente

Durante el desarrollo del frontend, se ha utilizado la tecnologia de Angular, puesto que es una tecnologia que actualmente se esta usando en profundida dentro del mundo web, al integrar el sistema de Modelo, Vista, Controlador, a su vez que a la vez utiliza HTML y CSS para dar el entorno visual. Con esta tecnología, se han usado los sigueintes modulos:

## Frameworks

Angular como se ha comentado, es un lenguaje que esta muy orientado a los entornos web, pero existen frameworks que dan soporte al desarrollo de aplicaciones moviles con esta tecnologia.

Existen varios frameworks en el mercado que pueden llevar a este resultado.

Uno de ellos es NativeScript, el cual se testeo durante el principio de este desarrollo y se comprobo que es un framework que aun sabiendo que esta en desarrollo, se encuentra falto de un montón de funcionalidades, a la vez que documentación para su desarrollo.

Tras comprobar que no era viable el desarrollo con dicho framework, se paso a usar **Ionic**, actualmente uno de los más utilizados dentro del mercado y que más caracteristicas y funcionalidades abarca.

Trabajando con él se ha comprobado que es bastante versatil y permite su desarrollo no solo desde entornos movil, sino que tambien permite el desarrollo dentro de cualquier navegador web. Lo que ofrece una alta capacidad para desarrollar dentro de cualquier entorno.

# Bibliográfica

Se añadirá una biografía más adelante.