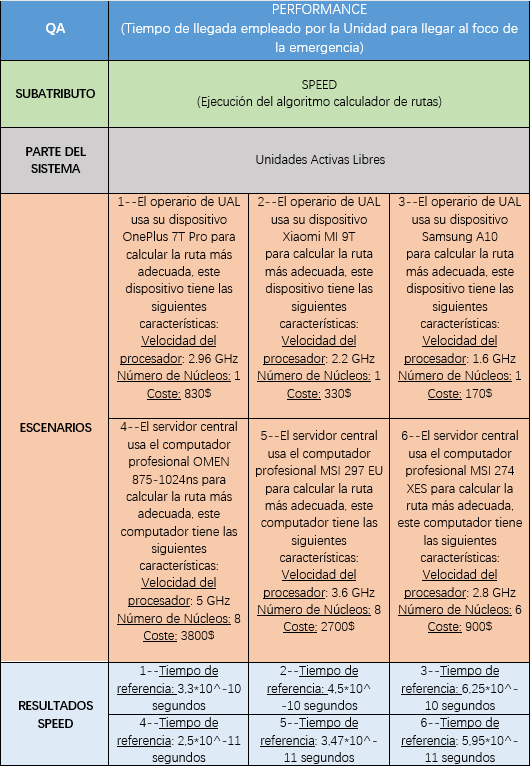
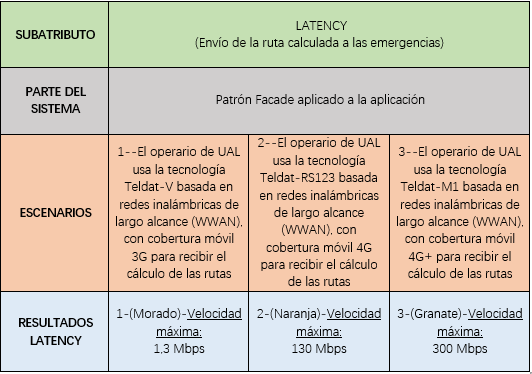
# QA PERFORMANCE

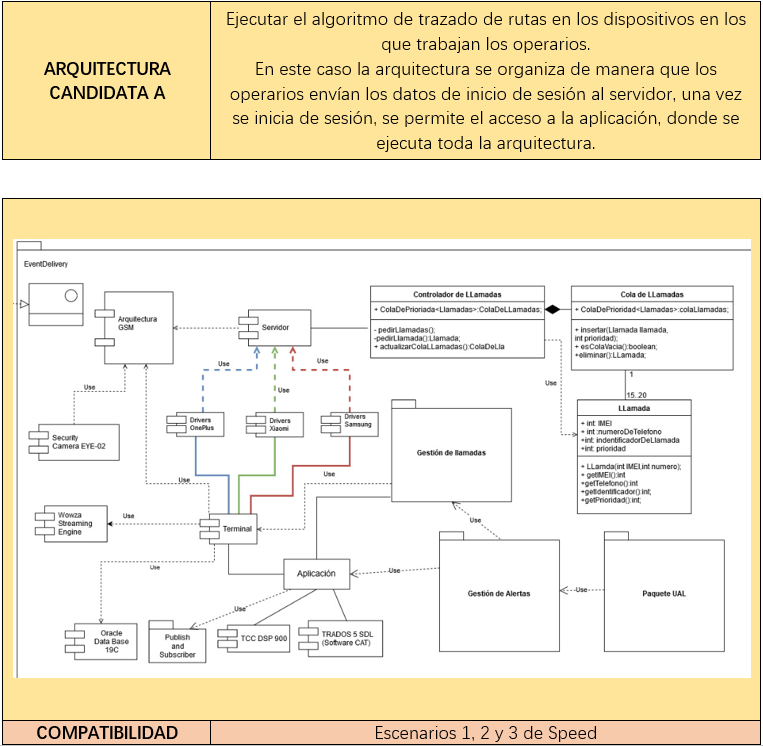
## UTILITY TREE EN FORMATO TABLA





PRIORIZACIÓN ESCENARIOS

EXPLICAR QUE CONTEMPLAMOS TODOS LOS ESCENARIOS EN DIFERENTES ARQUITECTURAS

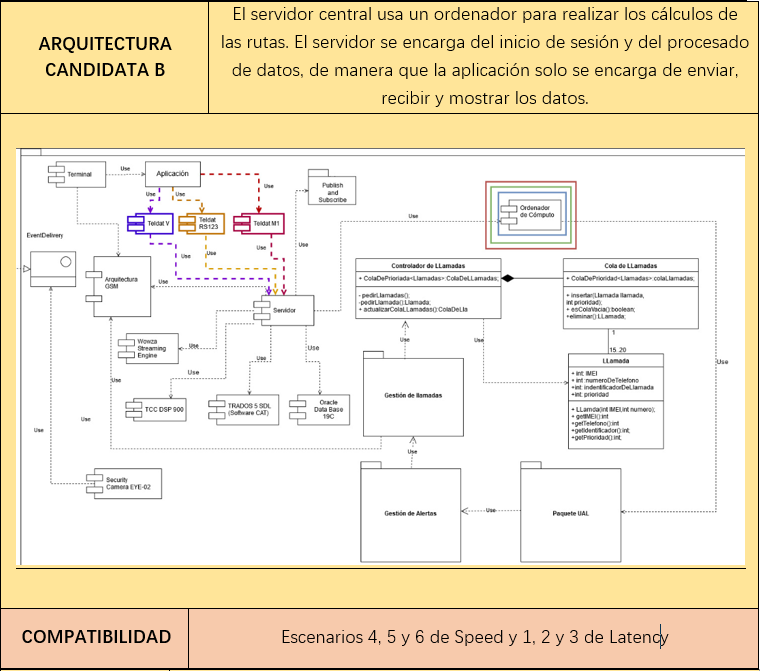


Sensitivity Points:

* Representar la separación entre servidor y aplicación.
* Reorganizar la comunicación entre subsistemas, ya que algunos dependerán de la aplicación y otros del servidor.

Riesgos:

* Se van ejecutar algoritmos que requieren altos recursos, y se está planteando hacerlo sobre teléfonos móviles, pudiendo hacerlo sobre otro tipo de equipos más potentes.
* La información esta menos centralizada, ya que hay subsistemas que no se comunican con el servidor.



Sensitivity Points:

* Representar que subsistemas se comunican con el ordenador de cálculo de rutas, que subsistemas se comunican con el servidor y que subsistemas se comunican con la aplicación.

Riesgos:

* La información está más centralizada en el servidor por lo que es más peligroso a la hora de que falle.
* La arquitectura es más dependiente de la comunicación con el servidor, ya que la aplicación no hace ningún tipo de cálculo, solo muestra la información que recibe del servidor.

Elección:

* Consideramos que la mejor candidata para nuestra arquitectura es la B. La elección es debido a que consideramos más adecuado tener un servidor centralizado, y que la aplicación en los terminales solo muestre la información. También la información estará mucho más centralizada, por lo que se podrá conocer en todo momento la posición de los operarios.

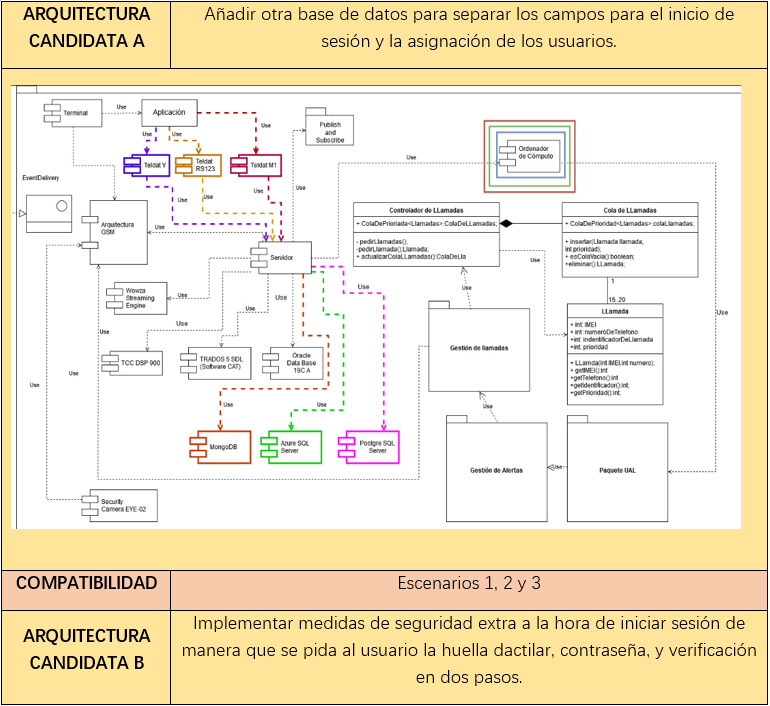
# QA SECURITY

## UTILITY TREE EN FORMATO TABLA



PRIORIZACIÓN ESCENARIOS

EXPLICAR QUE CONTEMPLAMOS TODOS LOS ESCENARIOS EN DIFERENTES ARQUITECTURAS

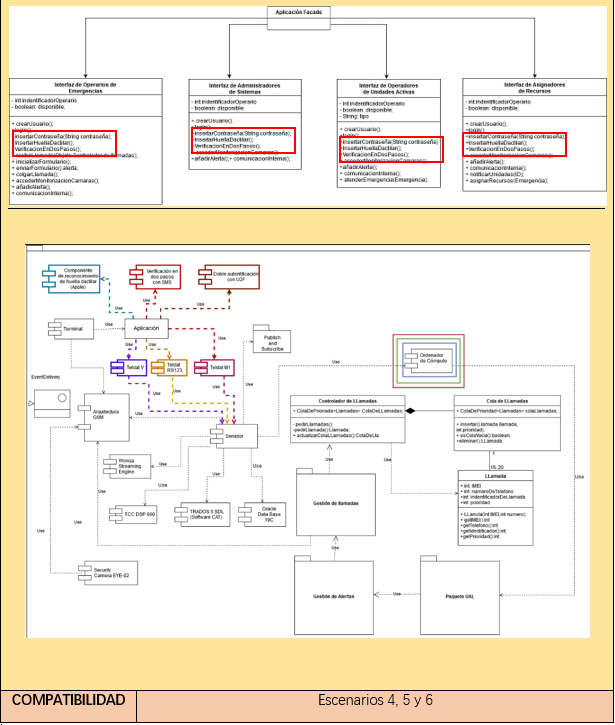


Sensitivity Points:

* Hay que modelar una segunda base de datos.

Riesgos:

* Hay que mantener otra Base de Datos.
* A parte de la contraseña, no hay otra medida de seguridad.



Sensitivity Points:

* Hay una gran cantidad de módulos para escoger a la hora de modelar nuestra arquitectura.
* Hay que modelar nuevas funcionalidades en la aplicación.

Riesgos:

* Hay menor seguridad en el inicio de sesión con las bases de datos.

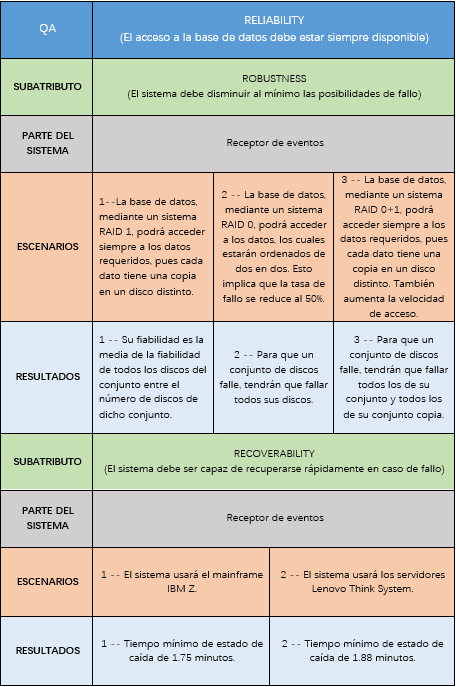
Elección:

* Consideramos más adecuada la elección de la arquitectura B frente a la A ya que debido a las características de nuestra arquitectura, consideramos que es más adecuado tener más seguridad al iniciar sesión que tener los datos divididos en dos bases de datos, ya que a nuestro sistema se puede acceder desde la aplicación desde terminales de los operarios.

Además de que creemos que es más improbable un ataque a una base de datos, que un ataque directo al inicio de sesión de la aplicación.

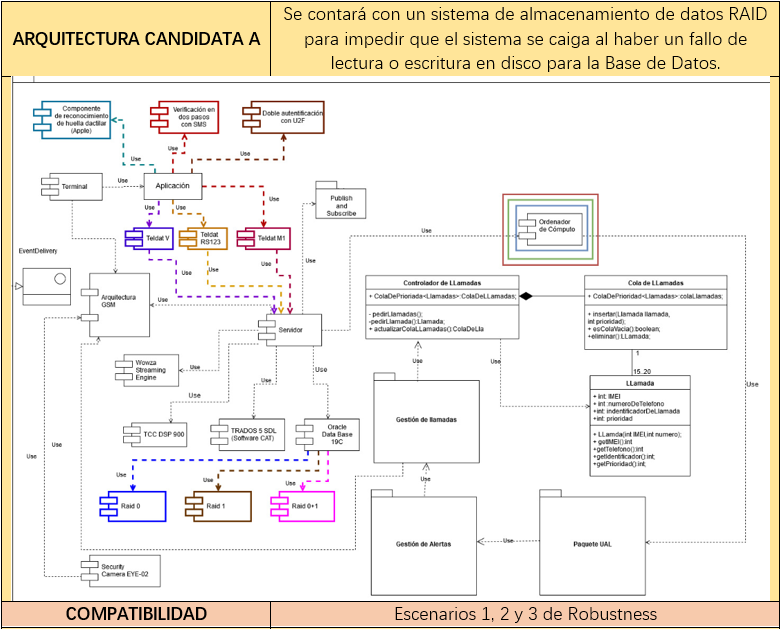
# QA RELIABILITY

## UTILITY TREE EN FORMATO TABLA



PRIORIZACIÓN ESCENARIOS

EXPLICAR QUE CONTEMPLAMOS TODOS LOS ESCENARIOS EN DIFERENTES ARQUITECTURAS

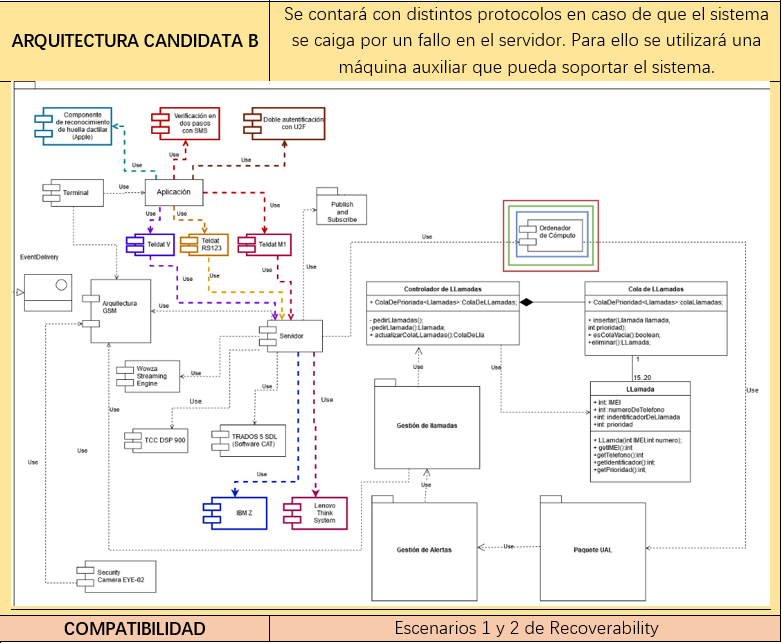


Sensitivity Points:

* Modelar las componentes que usa la base de datos para impedir que el sistema se caiga al haber un fallo.

Riesgos:

* Si ocurriese cualquier tipo de fallo, por improbable que sea, no hay ningún protocolo de recuperación.



Sensitivity Points:

* Cambiar la base de datos modelada por las bases de datos propuestas.

Riesgos:

* Tener una alta recuperación de fallos, no sirve de nada si nuestros servidores fallan mucho.

Evaluación:

* Consideramos más adecuada la elección de la arquitectura B frente a la A ya que debido a las características de nuestra arquitectura, consideramos que es más adecuado tener más seguridad al iniciar sesión que tener los datos divididos en dos bases de datos, ya que a nuestro sistema se puede acceder desde la aplicación desde terminales de los operarios.

Además de que creemos que es más improbable un ataque a una base de datos, que un ataque directo al inicio de sesión de la aplicación.