Requisitos no funcionales

Trivial5b - Versión 3

En la primera versión de la documentación se extraían los requisitos no funcionales que albergaban mayor importancia en nuestra primera parte del sistema, un sistema de extracción y transformación de datos con arquitectura Bash. En la segunda versión, aparecieron varios nuevos atributos del sistema ya que teníamos una aplicación de escritorio. Por último, en esta tercera y última entrega aparecerán nuevos requisitos relacionados con una aplicación web.

**Testabilidad**

Facilidad para probar y comprobar la fiabilidad del sistema, con garantía de que las preguntas han sido transformadas correctamente en ambas etapas del proceso, y que el juego funciona como se espera en todo momento.

**Integridad**

Los datos de la aplicación deben estar siempre consistentes, completos y correctos.

**Modularidad**

El sistema actual cuenta con varios módulos del sistema completamente distintos y los cuales son dependientes el uno del otro, por lo que han de coexistir entre ellos.

**Integración**

Es un atributo importante de nuestro sistema ya que se pretende integrar el sistema de extracción y transformación de datos en un nuevo sistema más complejo que comprende una aplicación de escritorio en primer lugar, y una aplicación web en segundo lugar. La integración de los datos abarca guardar los datos que una aplicación manipula de manera que otras aplicaciones puedan acceder a ellos.

**Modificabilidad**

El sistema ha de permitir la obtención de los datos de entrada con diferentes formatos por lo que se desea la posibilidad de incluir mayor funcionalidad en este aspecto en el futuro, al igual que ampliar la funcionalidad del juego en sí.

**Disponibilidad**

El operador encargado de la ejecución de las distintas etapas del proceso ha de ser capaz de iniciar cualquiera de ellas en cualquier momento que desee. En el caso del usuario, el juego debe de estar disponible para ser ejecutado en cualquier momento que este desee tanto en la aplicación de escritorio como en la aplicación web.

**Robustez**

Es la habilidad de un sistema para tratar los errores durante el proceso de ejecución. En nuestro caso sería, por ejemplo, el hecho de que en el proceso de tratamiento de los datos, tanto lectura como transformación, o a la hora de la ejecución de la aplicación pudiera existir algún error y el usuario no debe ser consciente de que esto ha ocurrido, de manera que el sistema trate el error y continúe con una ejecución normal.

**Usabilidad**

En la anterior versión, el operador encargado de la ejecución de las distintas etapas cuente con una interfaz sencilla y usable. En la actual versión, además, siendo tanto una aplicación de escritorio como web, cobra una mayor importancia este requisito ya que necesitamos que la interfaz del usuario sea sencilla e intuitiva.

**Portabilidad**

Se prevee que posteriormente se pueda crear una aplicación web similar a la actual aplicación de escritorio, por lo que pretendemos portar la aplicación a una diferente taxonomía.

**Accesibilidad**

La accesibilidad es la habilidad de un producto software de ser fácilmente accesible por una gran cantidad de aparatos tecnológicos y de distintos tipos de usuarios. Con esta tercera y última entrega se hace aún más importante, ya que sería más factible el hecho de ejecutar la aplicación desde cualquier dispositivo siendo esta una aplicación web.

**Escalabilidad**

Al implementar web una aplicación web, aparece este nuevo atributo. El sistema deberá diseñarse de forma que aunque en un inicio el número de usuarios sea bajo, si en un periodo corto de tiempo aumentasen, la disponibilidad y rendimiento de este no se vea afectado.

**Rendimiento**

Es un indicador de la capacidad de respuesta del sistema para ejecutar una acción dentro de un intervalo de tiempo dado. Puede ser medida en términos de latencia o de respuesta. En nuestro caso, en la última entrega, necesitamos que las acciones que se realicen dentro del juego sean lo suficientemente fluidas de manera que el usuario no note una ralentización en la ejecución.

**Seguridad**

Se trata de un requisito no funcional muy importante en una aplicación web. Es la forma en que un sistema es protegido de perder o suministrar información y la posibilidad de éxito de un ataque. Nuestro sistema trata de proteger los activos y previene la modificación de información de fuentes no autorizadas.