
Berserkers - RNF

- ★ Alejandro Mesa
 - ★ Cristhian González
 - ★ Hernán Aguilar
 - ★ Janer Torres
-

Introducción

La presente es una investigación sobre Requisitos No Funcionales (RNF) de un sistema de software, en la que pretendemos abordar algunos ejemplos, dando una definición, estableciendo métricas (en la medida de lo posible) y exponiendo diversos casos de uso para cada uno de ellos




RNF

Los siguientes son los **7 RNF** escogidos para esta investigación:

- **User-Friendliness**
- **Explotabilidad**
- **Integridad de datos**
- **Operabilidad**
- **Robustez**
- **Elasticidad**
- **Internationalization and localization**

User friendliness

User friendliness (O Facilidad de uso). Hace referencia a la capacidad de un software para ser aprendido, utilizado y a su vez ser amigable para un usuario, bajo ciertas condiciones.



Es esencial tenerlo en cuenta al momento de desarrollar una aplicación.

Requisitos

- **Atracción**
 - *Capacidad del software para ser amigable con el usuario. Hacer una aplicación más atractiva para el usuario.*
- **Facilidad de comprensión**
 - *Permitirle entender al usuario si su software es conveniente y cómo se pueden usar los servicios que este presta.*
- **Facilidad cognoscitiva**
 - *Hace referencia al tiempo de media que es necesario invertir para completar cada tarea.*
- **Operabilidad**
 - *Capacidad de ser controlado, la tolerancia ante errores y la conformidad con las expectativas del usuario*





Requerimientos

- **Control de errores**
 - *Presentar tipos de errores que se puedan dar, y una solución a estos para manejar de manera más fácil la aplicación.*
- **Interfaz amigable e intuitiva**
 - *Hacer del diseño algo entendible para todos, reduciendo la complejidad del programa y haciéndolo más visible.*
- **Información**
 - *Hacer uso de manuales que permitan entender la aplicación, como lo puede ser una documentación.*
- **Control de procesos**
 - *Identificar las tareas que hace la aplicación, para así poder terminarlas de manera satisfactoria sin que ocurran errores.*

Ejemplos

Los más usuales:

- Sistemas para gente mayor o un público no especializado
- Aplicaciones con objetivos específicos como:
 - Enseñar un idioma con facilidad
 - Aplicaciones de juegos para bebés
 - Interacción social



En general, es una regla que se busca seguir si se quiere comercializar un producto, por lo que lleva a una cantidad de ejemplos muy grandes.



¿Por qué?

“La definición de este requisito es una cuestión de especial importancia, pues un proyecto puede fracasar porque sus usuarios no encuentren sencillo su uso, porque no exista una interfaz sencilla y atractiva, o un manual que describa el funcionamiento y el uso del sistema al usuario final”.

Explotabilidad

Es la capacidad de que tiene un software para poder contrarrestar el uso o beneficio de su aplicación de tal manera que sea vulnerable

Explotabilidad

Se debe poder identificar debilidades en el sistema. Así mismo, saber si estas pueden ser explotables y que no exista una ruta por la cual esto se pueda hacer.

Si se realiza que existe alguna dificultad como esa, se deberá tomar medidas de seguridad adecuadas para que no exista algún riesgo.

Explotabilidad VS Vulnerabilidad

La explotabilidad es la encargada de encontrar estas vulnerabilidades. Así mismo, es capaz de evaluar su potencial.

“La vulnerabilidad se ocupa de lo teórico, la explotabilidad se ocupa de los datos reales”.



¿Por qué?

Una vez se tiene presente que pueden existir riesgos de seguridad en la aplicación, es evidente que se deben plantear soluciones para atacar estos mismos. Lo que presenta la explotabilidad es la manera en la cual actúan estas partes vulnerables del sistema.

Integridad de datos

Se refiere a la **precisión, completitud y coherencia** general de los datos.

Al momento de modificar los contenidos de una aplicación la integridad de los datos almacenados puede perderse de muchas maneras, esto se evita gracias a un conjunto de procesos, reglas y normas que se ponen en práctica durante la fase de diseño.



Este RNF fué descrito en el estándar internacional **ISO 25000** más específicamente **ISO 25010** para la evaluación de la **calidad del software**

Métricas

La integridad de los datos en un proyecto implica los siguientes aspectos:

- **Precisión de los datos**
 - *Que cada dato sea fiel representante de lo que requiere*
- **Confiabilidad de los datos**
 - *Dotando a la información de coherencia y estabilidad*
- **Completitud de los datos**
 - *Que se garantice que en ningún lugar falten campos ni valores*
- **Conformidad de los datos**
 - *Que se respete un formato determinado a la hora de ingresar datos*
- **Consistencia de los datos**
 - *Que los relaciona con las reglas de negocio existentes*



Tipos | Integridad

- **Integridad física**
 - *Catástrofes naturales, apagones o ataques a las funciones de una base de datos*
- **Integridad lógica**
 - *Protege los datos del error humano*
 - **Integridad de la entidad**
 - *Identificación única de datos mediante PK o valores únicos*
 - **Integridad referencial**
 - *Asegura que los datos se almacenen y utilicen uniformemente (FK)*
 - **Integridad de dominio**
 - *Valores aceptables que una columna puede contener. Puede incorporar restricciones*
 - **Integridad definida por el usuario**
 - *Reglas y restricciones creadas por el usuario para sus necesidades*



Qué no es Integridad de datos

➤ No es seguridad de datos

- *Si bien la integridad de los datos se ocupa de mantener la información intacta y veraz en el transcurso de su existencia, el objetivo de la seguridad es el de proteger la información de ataques externos.*

➤ No es calidad de datos

- *La calidad de datos determina si la base de datos cumple los estándares definidos por la empresa y las necesidades del negocio. La calidad al igual que la seguridad es tan solo una parte de la integridad de datos.*



Riesgos a la integridad

- Error humano
 - *Introducir información incorrecta*
- Errores de transferencia
 - *Los datos no alcanzan a llegar a su destino*
- Virus y errores de programación
 - *Estos son capaces de alterar, borrar o apropiarse de los datos*
- Hardware comprometido
 - *Problemas de funcionamiento de dispositivos*




Casos de Uso | ¡Contexto!

Este RNF se implementa directamente desde el diseño de la estructura del DBMS, ya sea con métodos tradicionales.

Pero en la actualidad la mejor opción es usar un proveedor de servicios desde el cual se ponen las aplicaciones a disposición de los usuarios a través de internet.

Gracias a herramientas de integración de datos en la nube, se pueden conectar múltiples aplicaciones y acceder a todos los datos de su empresa desde una **única ubicación**.



Una de las funciones importantes de un **DBMS relacional** es preservar la integridad de sus datos almacenados en la mayor medida posible.

Operabilidad

Capacidad de un sistema informático para hacer frente a los errores durante la ejecución y hacer frente a la entrada errónea. Robustez puede abarcar muchas áreas de la ciencia de la computación, tales como la programación sólida , robusta aprendizaje automático y robusta seguridad de la red .



Este RNF fué descrito inicialmente en el estándar internacional **ISO 9126** hoy conocido como **ISO 25000** para la evaluación de la **calidad del software**

Métricas

Como la 'Operabilidad' está contenida dentro de la 'Usabilidad' puede medirse con las 3 métricas fundamentales de la Usabilidad:

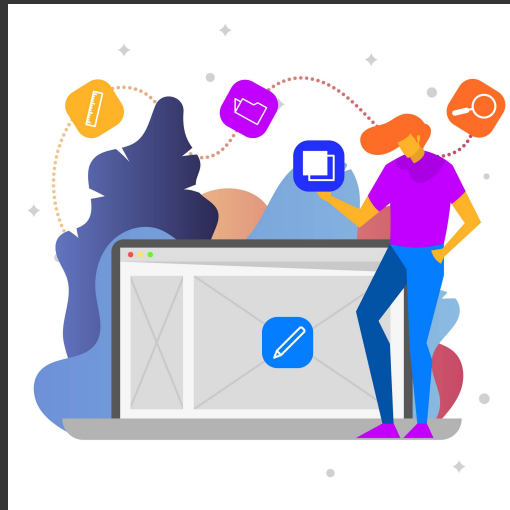
- **Medir la Efectividad**
 - *Hace referencia a los participantes que alcanzan de forma correcta cada objetivo. (Porcentaje de éxito)*
- **Medir la Eficiencia**
 - *Hace referencia al tiempo de media que es necesario invertir para completar cada tarea.*
- **Medir la Satisfacción**
 - *Puede medirse y calcularse por medio de la "Escala de Sistemas de Usabilidad".*



Casos de Uso | ¡Contexto!

Evidentemente todo aplicativo de software debe ser perfectamente operable, sin embargo, quienes deben procurar cumplir con este aspecto son aplicativos o servicios orientados al usuario (Páginas web, apps móviles, etc)

Es ideal que se implemente en la **Arquitectura Orientada a Servicios (SOA)** a través de Frameworks especializados en diseño UX/UI




Robustez

Capacidad de un sistema informático para hacer frente a los errores durante la ejecución y hacer frente a la entrada errónea.

Un programa es robusto si se comporta en forma razonable aún en circunstancias que no fueron anticipadas en la especificación de requerimientos



Este término ha sido reconocido por la **ANSI** y la **IEEE**



Un proceso es **robusto** si puede adaptarse a cambios no previstos en el entorno como ser una nueva liberación del sistema operativo o una transferencia repentina de la mitad de los empleados a otra sección.

Métricas

En este caso existen técnicas especializadas para medir la ‘Robustez’ de un sistema, las cuales se conocen como *Robustness testing*, una de las que más se destaca es el **Fuzzing** (o *Fuzz Testing*) que consiste en proporcionar datos inválidos, inesperados o aleatorios a las entradas del programa.

La **inyección de fallas** es una técnica para mejorar la cobertura de una prueba mediante la introducción de fallas en las rutas del código de prueba.



Casos de Uso | iContexto!

- Centrales de energía
- Centros de reclusión (Cárceles)
- Sistemas médicos
- Sistemas de Transporte
 - Transporte Autónomo (TESLA)
- MARES

Un sistema informático alcanza la *robustez* por el tipo de **programación** o por su **gestión de la información digital**.

Este RNF es perfectamente aplicable a cualquier tipo de patrón arquitectónico.



Elasticidad

Se define como "el grado en que un sistema es capaz de adaptarse a los cambios de la carga de trabajo mediante el aprovisionamiento y desaprovisionamiento de recursos de manera autónoma, de modo que en cada momento los recursos disponibles se ajusten lo más posible a la demanda actual".



Este RNF fué descrito inicialmente en la publicación **The NIST Definition of Cloud Computing** de **Peter Mell** y **Tim Grance** en 2011

Casos de Uso | iContexto!

Este RNF se aplica directamente a arquitecturas de computación en la nube y a día de hoy está ampliamente implementado en servicios cloud como:

- Amazon web services
- Microsoft Azure
- Google cloud services
- Heroku



Internacionalización

Localización

Son medios para adaptar software informático a diferentes idiomas, peculiaridades regionales y requisitos técnicos de un lugar de destino. La internacionalización es el proceso de diseñar una aplicación de software para que pueda adaptarse a varios idiomas y regiones sin cambios de ingeniería.

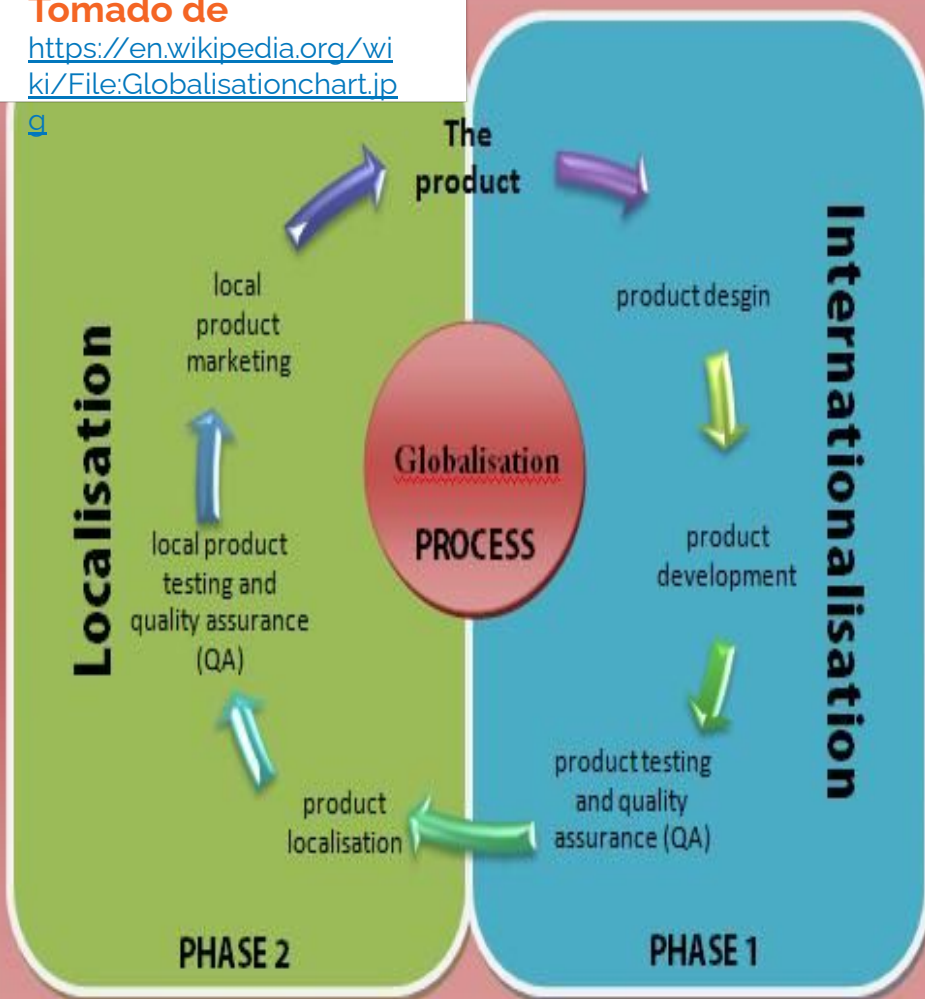


Este RNF fue descrito inicialmente en el libro **Software without frontiers: A multi-platform, multi-cultural, multi-nation approach** de **Patrick A.V. Hall** en 1997

Tomado de

<https://en.wikipedia.org/wiki/File:Globalisationchart.jpg>

g



Casos de Uso | iContexto!

Este RNF es altamente común encontrarlo en aplicaciones web o en software con distribución global, como por ejemplo:

- Facebook
- Windows 10
- Android

Este RNF también es compatible con cualquier patrón arquitectónico.