

**Integrantes:**

Barajas Diego

Cruz Brandonn

González Juan Sebastián

Miranda Nicolás

Orozco Juan David

*Healthy Routine:*

*Reporte plan de pruebas*

Pontificia Universidad Javeriana

Ingeniería de Sistemas

Ingeniería de Software



**Docente**: Anabel Montero

1. **INTRODUCCIÓN**

En el desarrollo de software es importante comprobar la fiabilidad y confiabilidad de un sistema, por ello se utilizan las pruebas de software. Estas son las que permitirán saber a los desarrolladores si el sistema cumple con los requisitos, o si presenta fallas de algún tipo. Es por este motivo que se presenta este reporte de pruebas de la aplicación Healthy Routine.

En este documento encontrará la descripción de la metodología utilizada para llevar a cabo las pruebas, el porqué se escogió y la forma como se adoptó. Finalmente, se presentará los resultados de las pruebas realizadas.

**2. DESCRIPCIÓN DE LA METODOLOGÍA**

Al momento de seleccionar la metodología para las pruebas de software, el equipo consideró buscar una que se ajustara al modo de trabajo que se ha seguido hasta el momento, por ello se consultó pruebas de software ágil. A continuación se describirán los principios y ventajas que ofrece Agile Testing*.*

**¿Qué es?**

“Agile Testing es una práctica de pruebas de software que sigue los principios del desarrollo ágil de software.” [1] Es decir que en esta metodología es más importante el trabajo iterativo que de una sola fase. Por tanto, las pruebas pasan a ser parte del desarrollo del proyecto, corrigiendo errores y dando retroalimentación. Esto a su vez permite una comunicación más fluida, que permitirá hacer correcciones en etapas más tempranas para reducir costos.

Otro de los enfoques de Agile Testing es el hecho de hacer que todos los integrantes participen en las pruebas. En otros métodos convencionales los testers realizan las pruebas en una determinada etapa del proyecto, pero en este caso al realizarse las pruebas en cada Sprint, los Analistas de negocio y Desarrolladores de software también ejecutarán las pruebas. Además, la metodología involucra al cliente en cada iteración.

Por último, ésta práctica reduce la documentación de las pruebas debido a que le da mayor importancia a la prueba en sí, en lugar de los detalles. Las principales herramientas usadas por los Agile Testers son las listas de checkeo reusables.

Algunas prácticas relacionadas de *Agile Testing* son:

* Testing guiado por pruebas (*Test Driven Development - TDD).*
* Desarrollo guiado por comportamiento *(Behavior Driven Development - BDD).*
* Desarrollo guiado por pruebas de aceptación *(Acceptance Test Driven Development - ATDD).*
* Testing exploratorio.
* Automatización de pruebas de regresión.
* Automatización de pruebas unitarias.

**2.1. Metodología seleccionada**

El formato seleccionado para llevar a cabo Agile Testing en el proyecto es el desarrollado por Briam Marick y luego adaptado por Lista Crispin y Janet Gregory en su obra “Agile Testing”, conocido como los cuadrantes de Agile Testing [1].

Esta herramienta ofrece diferentes tipos de pruebas ágiles para lograr diferentes objetivos en el proyecto. Estas pruebas se agruparán en cuatro cuadrantes, los cuales a su vez tienen un significado tanto para el eje horizontal como para el vertical. Por el lado del eje horizontal se encuentran las pruebas en apoyo al equipo *(Supporting the Team)* y las pruebas de crítica al producto *(Critique de Product).* Mientras que en el eje vertical se encuentran las pruebas de cara a la tecnología *(Technology Facing)* y de cara al negocio *(Bussiness Facing).*

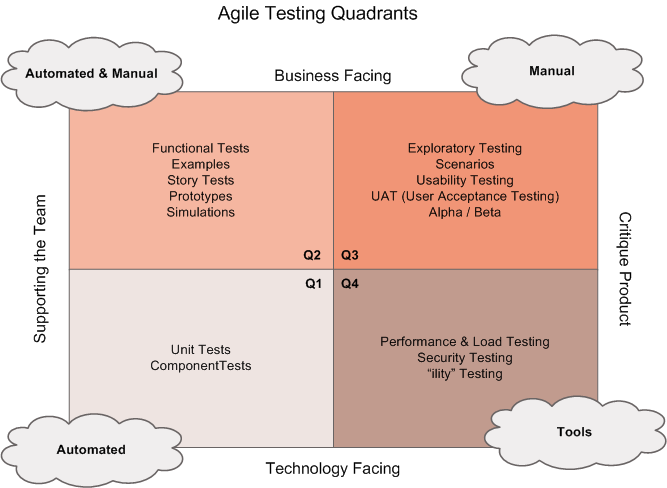
****

Fig 1. Agile Testing Quadrants

**Cuadrante 1: Pruebas de apoyo al equipo de cara a la tecnología**

Son el primer tipo de pruebas que se realiza sobre las funcionalidades implementadas. En este tipo de pruebas se encuentra:

* **Pruebas unitarias:** verificación de la funcionalidad de un pequeño subconjunto del sistema. Como se mostró en el SRS **(Ver anexo)** estas pruebas se llevan a cabo por el miembro al que le fue asignada la responsabilidad de implementar la funcionalidad. La plantilla que se utilizará para llevar el registro de las pruebas unitarias es:

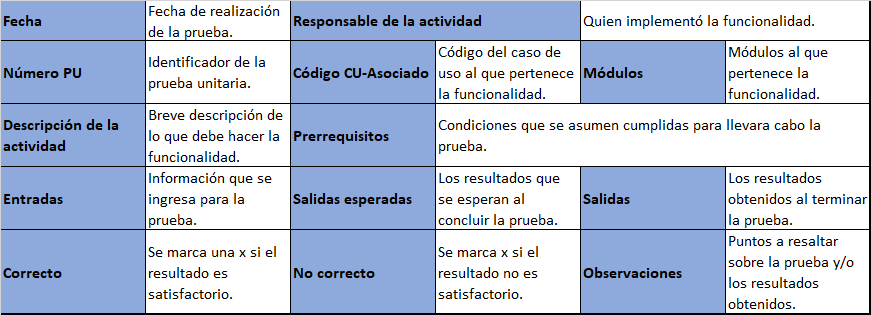


Fig 2. Plantilla para pruebas unitarias.

* **Pruebas de componente:** verificación del comportamiento de un grupo más grande del sistema. Estas pruebas pueden llevarse a cabo tanto por el responsable de la implementación de las funcionalidades o por algún otro miembro del grupo, ya que estos componentes pueden estar conformados por las funcionalidades implementadas por más de un miembro en un Sprint.

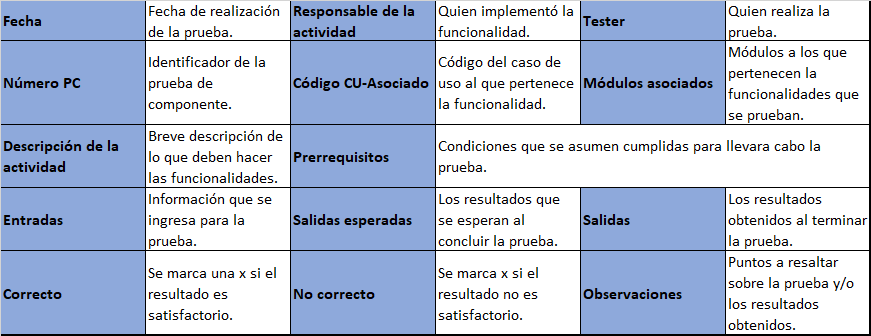


Fig 3. Plantilla para pruebas de componente.

**Cuadrante 2: Pruebas de apoyo al equipo de cara al negocio**

Son pruebas que permiten tener una visión más amplia del estado del producto, además, permiten comprobar la calidad que el cliente solicita. Estas pruebas se desarrollarán tanto en la parte lógica del producto como con interfaces.

Las características y actividades de estas pruebas son:

* Pruebas de posibles escenarios con flujos de trabajo proporcionados por el cliente.
* Prueba de par.
* Comprueban el correcto comportamiento del sistema (comprobación de los requisitos funcionales).

La plantilla que se utilizó para llevar el registro de estas pruebas es:

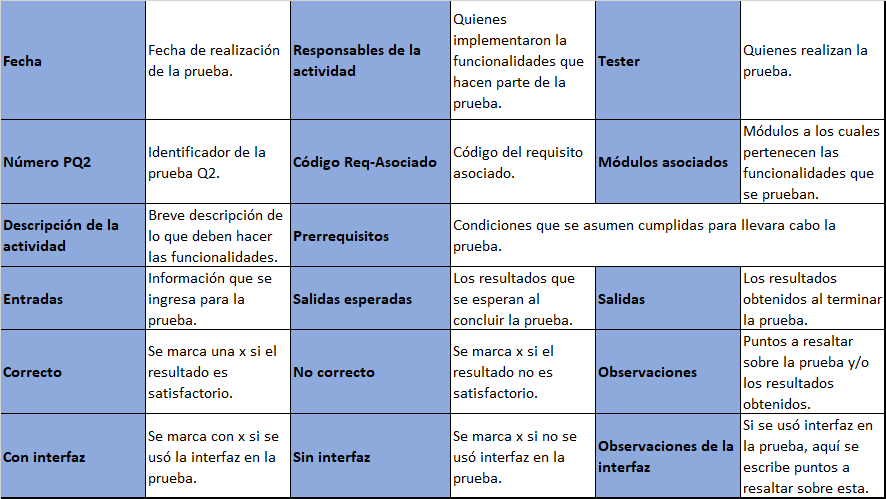


Fig 4. Plantilla para pruebas de Q2.

**Cuadrante 3: Pruebas que critican el producto de cara al negocio**

Pruebas con usuarios finales.

* Pruebas de usabilidad.
* Pruebas exploratorias.
* Pruebas de pares con clientes.
* Pruebas colaborativas.
* Pruebas de aceptación del usuario.

**Cuadrante 4: Pruebas que critican el producto**

En este cuadrante se encuentran las pruebas que permiten la verificación del cumplimiento de los requisitos no funcionales. Algunas de estas son:

* Pruebas no funcionales como pruebas de estrés y rendimiento.
* Pruebas de seguridad con respecto a la autenticación y piratería.
* Pruebas de infraestructura.
* Pruebas de migración de datos.
* Pruebas de escalabilidad.
* Prueba de carga.

Las pruebas escogidas por el grupo para este cuadrante son: pruebas de rendimiento, persistencia, escalabilidad, de borde y soporte.

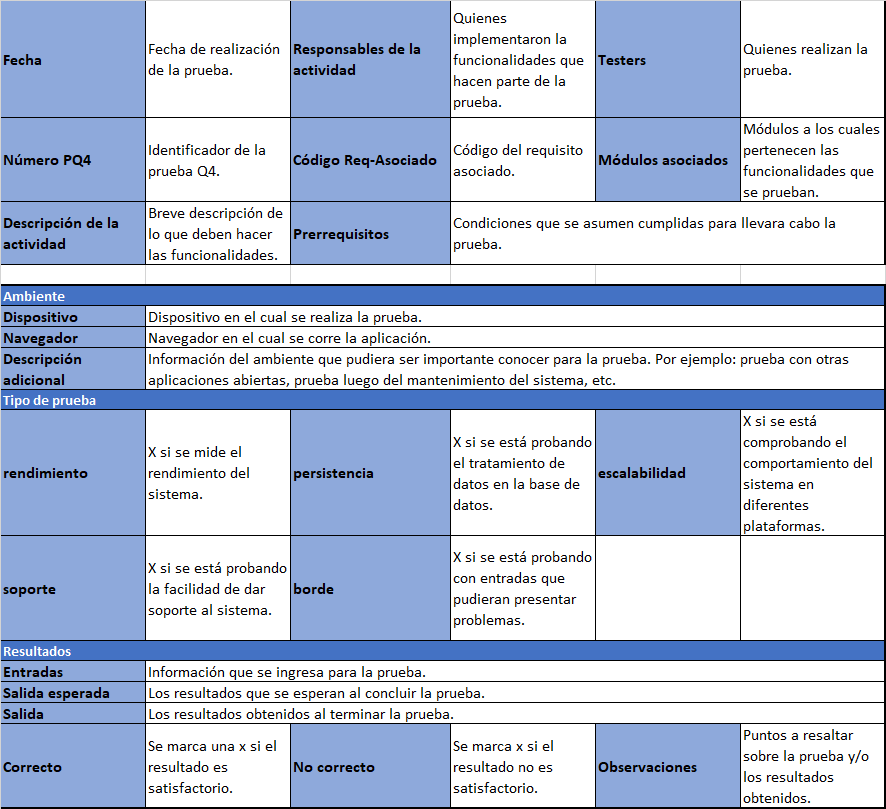


Fig 5. Plantilla para pruebas de Q3.

**REFERENCIAS**

[1] PMOinformática, *“Qué es el Agile Testing y cuáles son sus principios y estrategias”,* 2015. [Online]. Available: <http://www.pmoinformatica.com/2015/04/pruebas-software-agile-cuadrantes-1.html>

[2] V. Loaiza y L. Zorro. *“Plan de Pruebas de Software STP“,* 2010. [Online]. Avalible: <http://pegasus.javeriana.edu.co/~CIS1010IS05/Documentos/Dise%C3%B1o/STP.pdf>

IEEE%20829-2008

<http://pegasus.javeriana.edu.co/~CIS1230TK04/Documentos%20P%C3%A1gina%20Pegasus/STP%20-%20DiaFoot2.pdf>

<http://pegasus.javeriana.edu.co/~CIS1510AP02/Documentos/5.0_Pruebas/5.0_STP_Documento_Pruebas.pdf>