



**Automatización de pruebas para el aplicativo FactoryTalk Hub de la compañía Rockwell  
Automation**

Juan Esteban García Muñoz

Trabajo de prácticas presentado como requisito parcial para optar al título de:  
Ingeniero de Sistemas

Carlos Mario Sierra Duque, Doctor en Ingeniería de Sistemas

Universidad de Antioquia  
Facultad de Ingeniería  
Ingeniería de Sistemas UdeA  
Medellín  
2023



## Referencia

- [1] J. E. García Muñoz, “Automatización de pruebas para el aplicativo FactoryTalk Hub de la compañía Rockwell Automation”, Presencial, Ingeniería de Sistemas, Universidad de Antioquia, Medellín, 2023.

Estilo IEEE (2020)



**Repositorio Institucional:** <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - [www.udea.edu.co](http://www.udea.edu.co)

**Rector:** John Jairo Arboleda Céspedes.

**Decano/Director:** Julio César Saldarriaga

**Jefe departamento:** Diego Jose Luis Botia Valderrama.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

## **Agradecimientos**

Inicialmente gracias a mi familia por todo el apoyo y la educación que me han brindado durante todo este tiempo, su esfuerzo, tiempo y dedicación han sido mi mayor motivación para llegar cada vez más lejos. Gracias igualmente a mi alma mater, allí descubrí formas diferentes de percibir y pensar sobre este mundo, lo que me rodea y apasiona. De todas las opciones posibles para emprender mi camino como profesional fue la mejor que pude haber escogido, viviré eternamente agradecido con la Universidad de Antioquia por brindarme tantos conocimientos y abrirme puertas que ni siquiera yo imaginaba. Allí entendí el valor de dar siempre lo mejor de sí, de ser disciplinado, de nunca renunciar y de siempre tener metas y objetivos en todo momento.

A todos mis profesores, compañeros, líderes y amigos que me brindaron su apoyo y conocimiento les agradezco inmensamente, sin ellos, hubiera sido prácticamente imposible lograr avanzar por este camino.

Finalmente, estoy muy agradecido con las empresas que han creído en mí, cuyas enseñanzas, experiencias, amigos serán siempre importantes para mi desarrollo profesional. Le agradezco a Rockwell Automation, donde actualmente ejerzo como Test Automation Engineer por permitirme ingresar a una empresa gigante, fabulosa, llena de profesionales que me inspiran a ser cada día una mejor persona y mediante la cual evidencie el gran valor de haber sido disciplinado.

## TABLA DE CONTENIDO

RESUMEN	8
ABSTRACT	8
I. INTRODUCCIÓN	9
II. OBJETIVOS	11
A. Objetivo general	11
B. Objetivos específicos	11
III. MARCO TEÓRICO	11
IV. METODOLOGÍA	15
V. RESULTADOS	23
VI. CONCLUSIONES	26
REFERENCIAS	27
VII. ANEXOS	28

## LISTA DE TABLAS

TABLA I: MÉTRICAS DE AUTOMATIZACIÓN POR SPRINT	27
--	----

## LISTA DE FIGURAS

Fig. 1.Ciclo de vida del desarrollo de software	12
Fig. 2.Servicios disponibles	13
Fig. 3.Menú de opciones	14
Fig. 4.Metodología SCRUM	16
Fig. 5.Repositorio de GitHub del proyecto para las pruebas	16
Fig. 6.Qtest consolidado de Sprints	17
Fig. 7.Encarpetado interno de QTest	17
Fig. 8.Composición interna de las carpetas en cada Sprint	19
Fig. 9.Estructura del código del proyecto.	20
Fig. 10.Estructura interna de las pruebas funcionales	21
Fig. 11.Archivos para la ejecución de pruebas	22
Fig. 12.Carpeta src	23
Fig. 13.Evolución total casos automatizados vs cobertura	24
Fig. 14.Cantidad de anomalías reportadas por Sprint	25

## SIGLAS, ACRÓNIMOS Y ABREVIATURAS

<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>Cms.</b>	Centímetros
<b>ERIC</b>	Education Resources Information Center
<b>Esp.</b>	Especialista
<b>MP</b>	Magistrado Ponente
<b>MSc</b>	Magister Scientiae
<b>Párr.</b>	Párrafo
<b>PhD</b>	Philosophiae Doctor
<b>PBQ-SF</b>	Personality Belief Questionnaire Short Form
<b>PostDoc</b>	PostDoctor
<b>UdeA</b>	Universidad de Antioquia



## RESUMEN

En este documento se presenta el trabajo realizado durante el periodo de prácticas académicas en el cual se propuso desarrollar pruebas automatizadas para el software FactoryTalk Hub, componente desarrollado en el proyecto Common Services and Unified Experience de la empresa Rockwell Automation, haciendo uso de herramientas de automatización de pruebas e integrando estas con el ciclo de vida del desarrollo del software, buscando mejorar la eficiencia de los procesos de pruebas durante la constante evolución del proyecto. Se explican los conceptos relacionados con este proceso de entre los cuales se encuentran, DevOps, CI/CD, Gherkin y de forma general cómo se implementaron las pruebas trabajando en una metodología ágil llamada Scrum, con el framework Webdriver IO, bajo el patrón POM en JavaScript, utilizando un repositorio para el control de versiones, además de otros conocimientos adquiridos a raíz del trabajo como la configuración de clusters para ejecutar las pruebas automatizadas en la nube. Finalmente, se resalta que además de cumplir con los objetivos planteados al inicio de las prácticas y adquirir conocimientos, llevar a cabo esta etapa ha permitido apoyar el crecimiento de los procesos de automatización y pruebas dentro de la compañía.

***Palabras clave* — Pruebas de software , Calidad de software, Automatización de pruebas, Control de calidad**

## ABSTRACT

This document presents the work carried out during the academic internship period in which automated tests were created for the FactoryTalk Hub software, a component developed in the Common Services and Unified Experience project of the Rockwell Automation company, using automation tools tests and integrating these with the software development life cycle, seeking to improve the efficiency of the testing processes during the constant evolution of the project. The concepts related to this process are explained, among which are DevOps, CI/CD, Gherkin and in a general way how tests were implemented in an agile methodology called Scrum, with the Webdriver IO framework, under the POM pattern in JavaScript. , using a repository for version control, in addition to other knowledge acquired as a result of the work such as configuring clusters to run automated tests in the cloud. Finally, it is highlighted that in addition

to achieve the objectives set at the beginning of the practices and acquiring knowledge, carrying out this stage has allowed the growth of automation and testing processes within the company.

**Keywords — Software Testing, Software Quality, Testing Automation, Quality Assurance**

## I. INTRODUCCIÓN

En Rockwell Automation, la asesoría y prestación de servicios a otras organizaciones para la automatización de sus procesos industriales y de manufactura son la base de la compañía, sin embargo, un gran reto era poder ofrecer a los posibles clientes la facilidad de administrar, conocer y utilizar los diferentes servicios que la compañía ofrece en los distintos sectores de automatización industrial. Los productos ofrecidos por la compañía se dividen principalmente en tres pilares, Design, Operations & Maintenance. Cada uno con un enfoque específico, Design se especializa en el diseño de automatización industrial en la nube y está pensado para grandes empresas de automatización industrial. Por otra parte, Operations tiene como fin, el control sobre la producción, ejecución de manufactura, puntos de control y líneas de producción de una compañía. Finalmente, Maintenance, está enfocado a la gestión de productos de maquinaria industrial, mantenimiento preventivo y correctivo. De allí, se desprende una gran cantidad de servicios específicos como: Arena™, Optix™, Studio 5000™, Remote Access™, Plex™ y Fiix™, etc, cada uno enmarcado dentro de alguno de los 3 pilares anteriormente mencionados.

De allí surgió la iniciativa de crear un aplicativo como FactoryTalk Hub (FTHub), ya que, a través de la misma, los clientes de la empresa Rockwell Automation pueden tener de manera centralizada el acceso a los servicios anteriormente mencionados y pueden estar en contacto remotamente con sus organizaciones y realizar una serie de tareas, tales como:

- Asignar los créditos (ver referencia en marco teórico) que la organización tenga disponibles en la cantidad y diversidad de servicios que se deseen.
- Monitorear los servicios, clientes u otras empresas a los cuales la organización tiene acceso.
- Editar la información de su organización.
- Invitar a otros clientes o empresas a hacer parte de la organización.

Puesto que todo ciclo de vida en el desarrollo de software incluye una etapa de pruebas para el aseguramiento de la calidad la compañía adoptó un área especializada para cumplir con dicha función. En el presente documento se ilustra la metodología de pruebas empleadas, las tecnologías, avances y procesos involucrados durante todo el periodo de prácticas cuyo fin era cumplir el objetivo de crear pruebas automatizadas para agregar valor y asegurar la calidad al momento de la entrega, despliegue y mantenimiento del software FactoryTalk Hub.

Es importante mencionar que debido a contratos de confidencialidad de la información y datos del cliente para el que se ha trabajado durante las prácticas académicas, no se puede compartir información con un alto nivel de detalle, por lo cual en algunos temas se habla de una forma más general y en otros se mostrarán ejemplos que permitan ilustrar algunos de los resultados obtenidos.

## II. OBJETIVOS

### *A. Objetivo general*

Desarrollar pruebas automatizadas para el software FactoryTalk Hub, componente desarrollado en el proyecto Common Services and Unified Experience de la empresa Rockwell Automation, con el propósito de agregarle valor y asegurar su calidad al momento de la entrega, y durante su despliegue y mantenimiento.

### *B. Objetivos específicos*

- Mantener y diseñar scripts automatizados para pruebas por interfaz y por servicio.
- Revisar y/o detectar anomalías presentes en el software del proyecto.
- Incrementar las métricas de automatización incluyendo escenarios de prueba de las funcionalidades nuevas que son entregadas en cada sprint y manteniendo las previamente entregadas.

## III. MARCO TEÓRICO

FactoryTalk Hub se refiere a un software creado por la compañía Rockwell Automation en el proyecto Common Services And Unified Experience con el fin de facilitar tareas que ayudan a los clientes de Rockwell Automation a ser más eficientes ya que tienen a disposición una plataforma que les permite estar conectados directamente con los servicios y diferentes dispositivos involucrados en cada proceso, además de poseer de manera centralizada la información de interés para los mismos, tal y como lo son las diferentes organizaciones y sus respectivas características (servicios disponibles, usuarios invitados, balance de créditos, entre otros).

Tal aplicativo, requiere de un proceso de pruebas el cual hace parte de todo ciclo de desarrollo de software, para de esta manera asegurar la calidad del mismo, y poder brindar finalmente un servicio de calidad a los usuarios.



Figura 1. Ciclo de vida del desarrollo de software

Nota: fuente <https://blog.innevo.com/calidad-en-el-desarrollo-de-software>

Las pruebas desempeñan un papel fundamental en el ciclo de desarrollo del software por varias razones de gran importancia

- **Identificación de Errores:** Las pruebas permiten detectar errores y defectos en el software antes de que llegue a los usuarios finales. Esto ayuda a corregir los problemas presentes en el aplicativo y asegura que el software funcione correctamente y cumpla con los requisitos.

- **Aseguramiento de Calidad:** Las pruebas ayudan a garantizar la calidad del software. Al evaluar su funcionamiento y desempeño en diferentes situaciones, se asegura que cumpla con los estándares y expectativas de los usuarios. Un factor muy importante a la hora de desplegar y entregar el producto a los usuarios de la compañía.
- **Reducción de Costos:** Encontrar y corregir errores durante las etapas de pruebas es mucho más económico que hacerlo después del lanzamiento. Esto evita gastos adicionales y problemas graves que puedan surgir en producción. [9]
- **Mejora de la Experiencia del Usuario:** Las pruebas aseguran que el software sea intuitivo, funcione sin problemas y cumpla con las necesidades y expectativas de los usuarios. Esto conduce a una experiencia positiva y satisfactoria para los usuarios. [9]

Razones por las que la etapa de pruebas en cualquier proyecto sea de ámbito de software, o no, es tan importante, pues la calidad del producto final depende de ello.

A continuación, se ilustrara de manera muy general el aplicativo FactoryTalk Hub, teniendo en cuenta los límites de confidencialidad respectivos. Es importante, primero, tener en cuenta algunos conceptos claves para su entendimiento y posteriormente, cómo fue llevado a cabo el proceso de pruebas (Ver glosario de términos en la sección de Anexos).

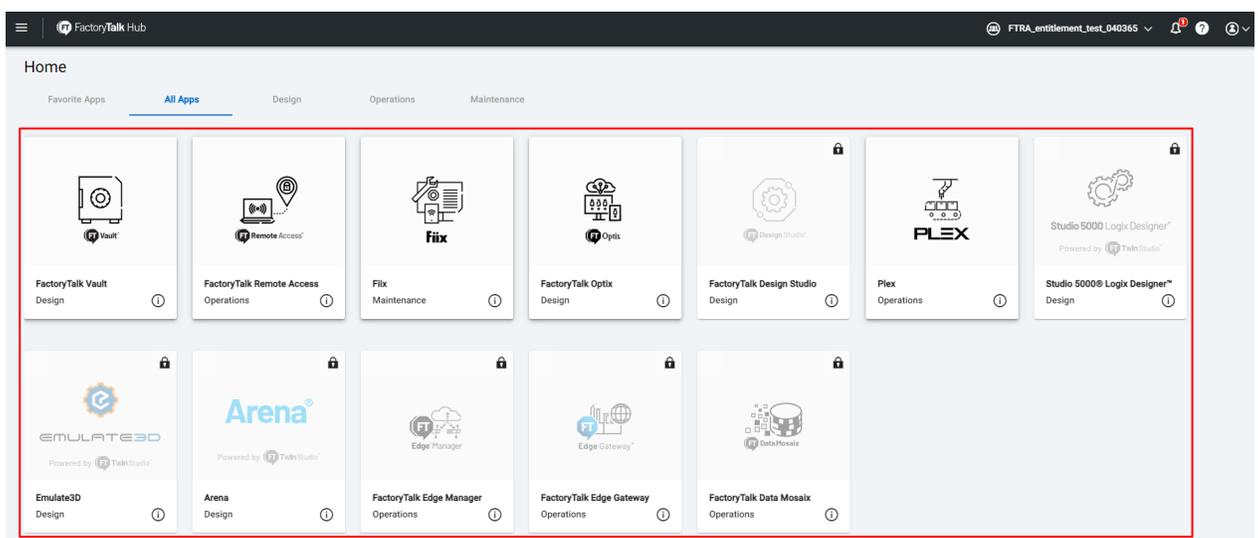


Figura 2. Servicios disponibles

En la figura 2, se enseña la página principal del aplicativo donde se aprecian los servicios a los cuales el cliente/usuario puede acceder a través del portal (FactoryTalk Hub).

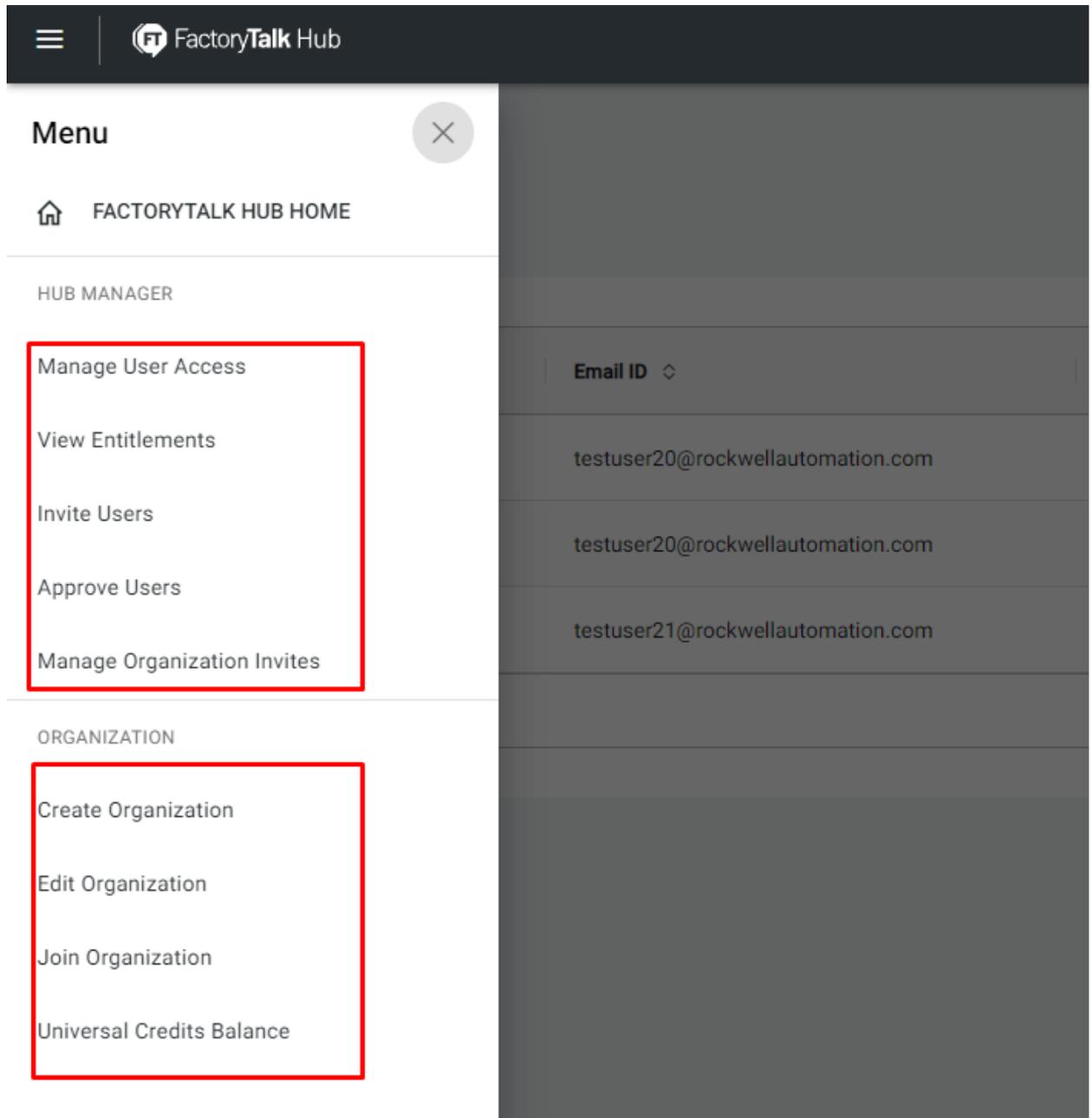


Figura 3. Menú de opciones

El menú del aplicativo se divide en dos secciones (Hub manager y Organization) respectivamente de acuerdo al enfoque de funcionalidades. En la primera sección, se encuentran las opciones cuyos fines están orientados a la administración de servicios (ver figura 3),

entitlements, interacción con demás usuarios u organizaciones y el manejo de dichos accesos. Por otra parte la segunda sección, está orientada a la administración de las organizaciones, desde allí se pueden crear o editar las mismas, además, de tener un consolidado de los créditos.

Una de las funciones principales del aplicativo es poder adquirir un entitlement para poder hacer uso de un determinado servicio o compra de créditos. Se tiene una lista de todos los entitlements que el usuario/cliente ha adquirido a lo largo del tiempo, el número de contrato, la fecha de adquisición/expiración respectivamente, el estado del contrato y la opción de asignar (provisionar) dicho entitlement.

Se tiene además una tabla donde se guardan los registros respecto al gasto y/o compra de los créditos. De esta forma, el usuario/cliente puede tener de manera clara y precisa información relevante respecto a la trazabilidad de sus créditos e incluso obtener informes de los mismos por ciertos periodos de tiempo.

Finalmente, otra de las principales características de la aplicación es la administración de los accesos concedidos a otros usuarios/clientes dentro de una organización en específico, donde el dueño de la organización puede editar o incluso eliminar cualquiera de los accesos dados a demás usuarios/clientes.

#### IV. METODOLOGÍA

Durante todo el proceso, se trabajó bajo una metodología ágil, en la cual se tenían sesiones diarias donde se informaba el avance del día anterior, lo que se iba a realizar y si se tenía algún impedimento. También se tenían reuniones al finalizar cada Sprint donde se hacían entregas de valor, retrospectivas para hacer retroalimentación con el equipo e incluso refinamientos para trabajar sobre las historias de usuario que serían tomadas posteriormente, llevando estricto seguimiento de la metodología ágil SCRUM [8].

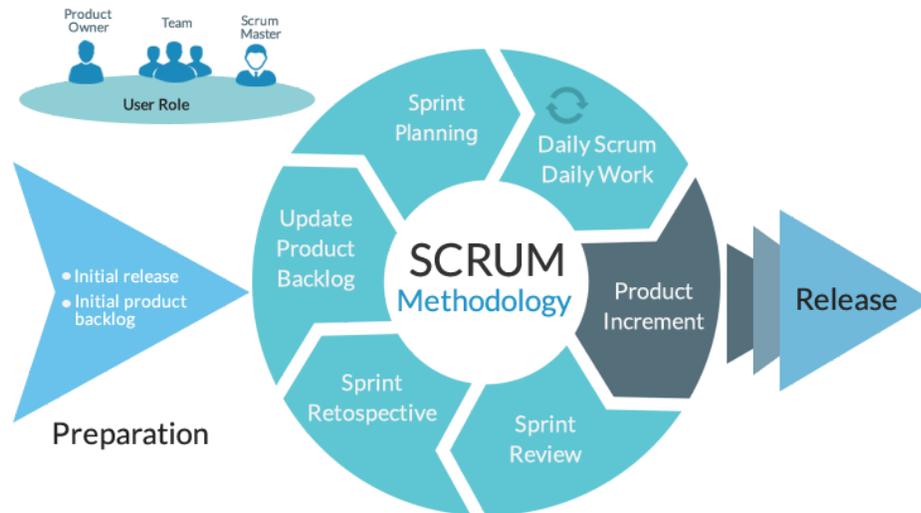


Figura 4. Metodología SCRUM

Nota: fuente: <https://blog.wearedrew.co/productividad/-ventajas-y-desventajas-de-la-metodologia-scrum>

Durante todo el desarrollo del proyecto a nivel de pruebas, se ha contado con un repositorio y un controlador de versiones [6], mediante el cual el equipo de pruebas realiza sus aportes personales sin afectar los cambios realizados por los demás integrantes del equipo a nivel del código.

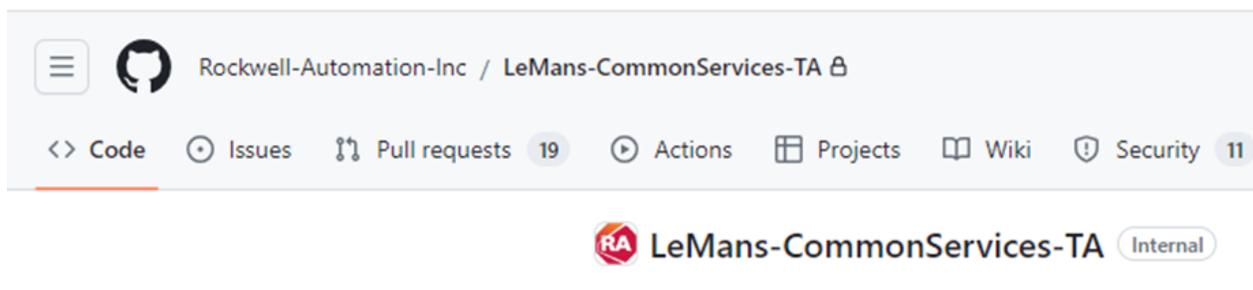


Figura 5. Repositorio de GitHub del proyecto para las pruebas

Adicionalmente, se usó una herramienta llamada QTest, mediante la cual el equipo podía evidenciar los resultados de la ejecución de las pruebas en la nube, orquestar los despliegues en producción, realizar pruebas de regresión entre otras actividades, todo gracias a la conexión de esta herramienta con las historias de usuario creadas en Jira[7].

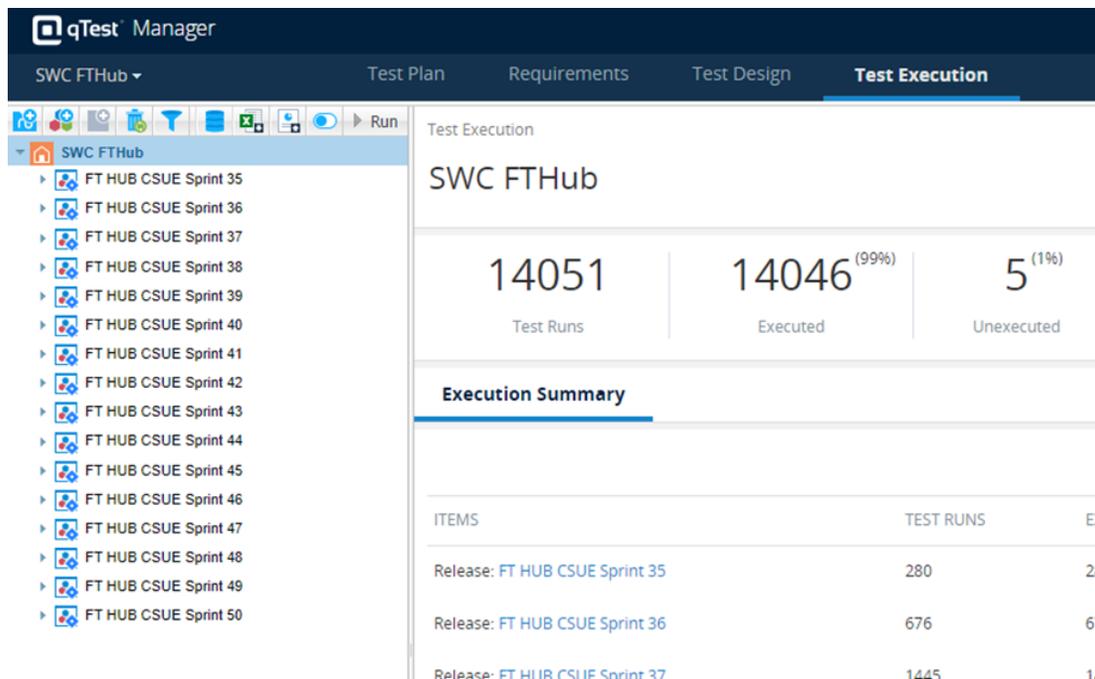


Figura 6. Qtest consolidado de Sprints

QTest a través de su sección de Test Execution permite ver todos los resultados de las pruebas ejecutadas en cada uno de los Sprint o interacciones del proyecto.

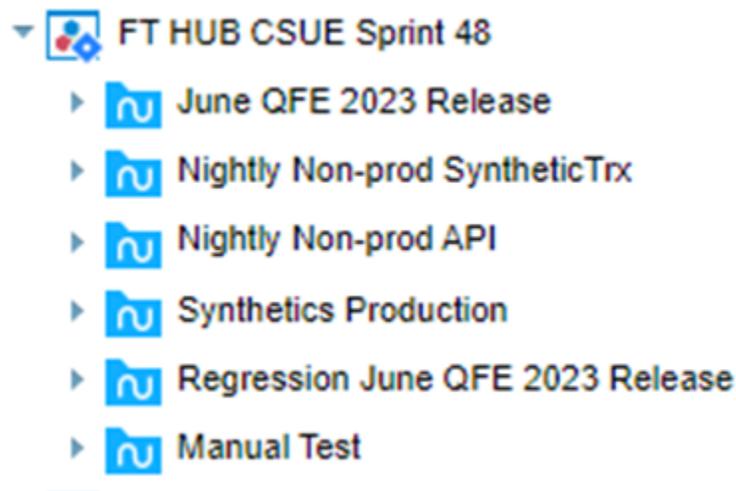


Figura 7. Encarpetado interno de QTest

En cada Sprint se tenían siempre 3 carpetas (Nightly Non-prod SyntheticTrx, Nightly Non-prod API y Synthetics Production) donde se visualizaban los resultados de las diferentes pruebas por funcionalidad. Sin embargo, en Sprints en los cuales habían despliegues, tal y como ocurrió en el Sprint 48, se creaban 3 carpetas adicionales cuyo objetivo era el siguiente respectivamente:

- June QFE 2023 Release: Contenía los resultados de las pruebas de las funcionalidades agregadas en ese respectivo periodo de tiempo (Junio) para dicho despliegue.
- Regression June QFE 2023 Release: Contenía los resultados de la prueba de regresión para dicho despliegue, teniendo en cuenta que, una prueba de regresión tiene como objetivo verificar que las funcionalidades antiguas no sean afectadas por la inclusión de nuevas al aplicativo.
- Manual Test: Esta carpeta se usaba para los resultados de las pruebas de confirmación de las anomalías encontradas durante el Sprint o despliegue.

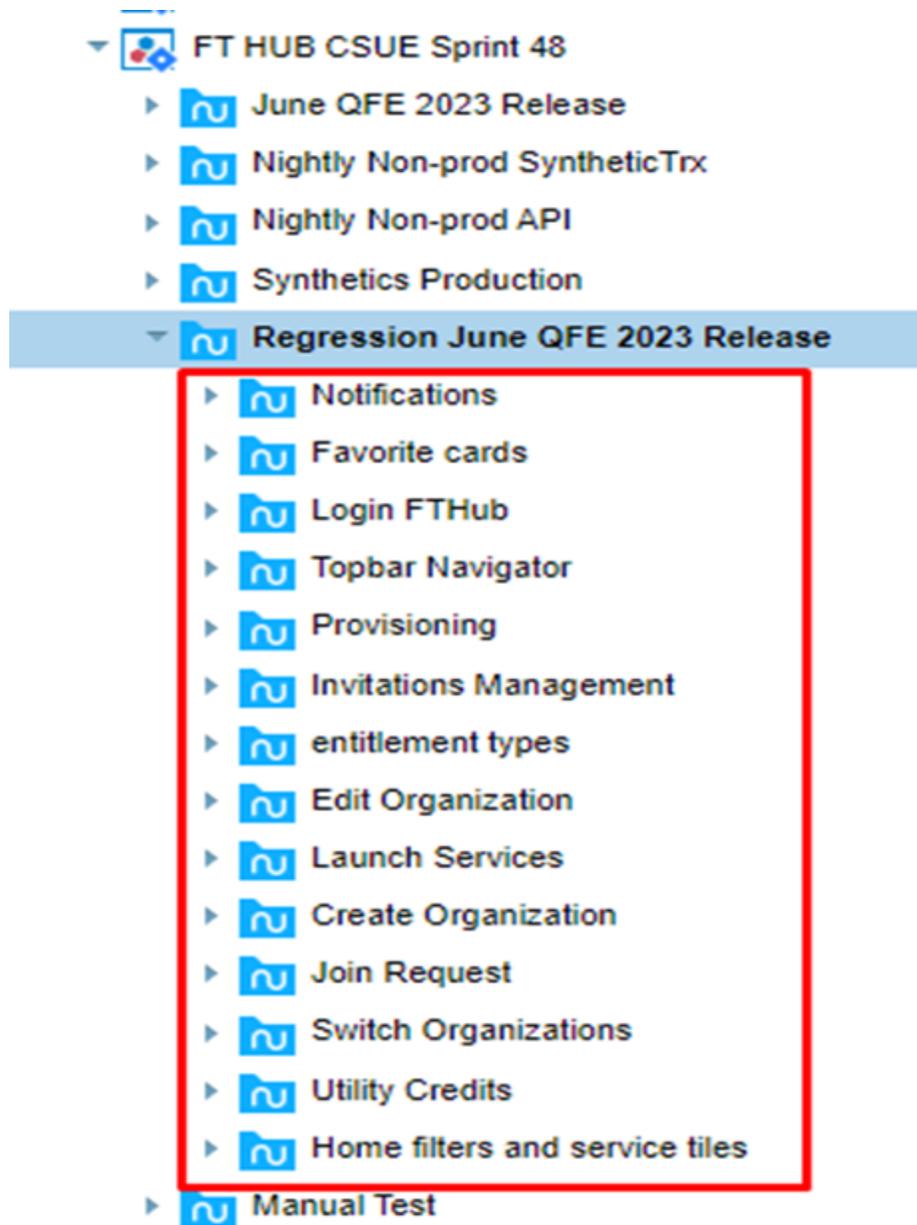


Figura 8. Composición interna de las carpetas en cada Sprint

Las carpetas al interior se encuentran compuestas por subcarpetas correspondientes a las diferentes funcionalidades del aplicativo, de esta forma se tiene una mayor organización de los casos de prueba diversificados por funcionalidad y facilita la búsqueda de los mismos dentro de la herramienta QTest.

El proyecto como se menciona al inicio del documento, fue trabajado a través del framework de WebDriverIO usando el lenguaje de JavaScript con el patrón de diseño POM, lo cual se detalla a continuación:

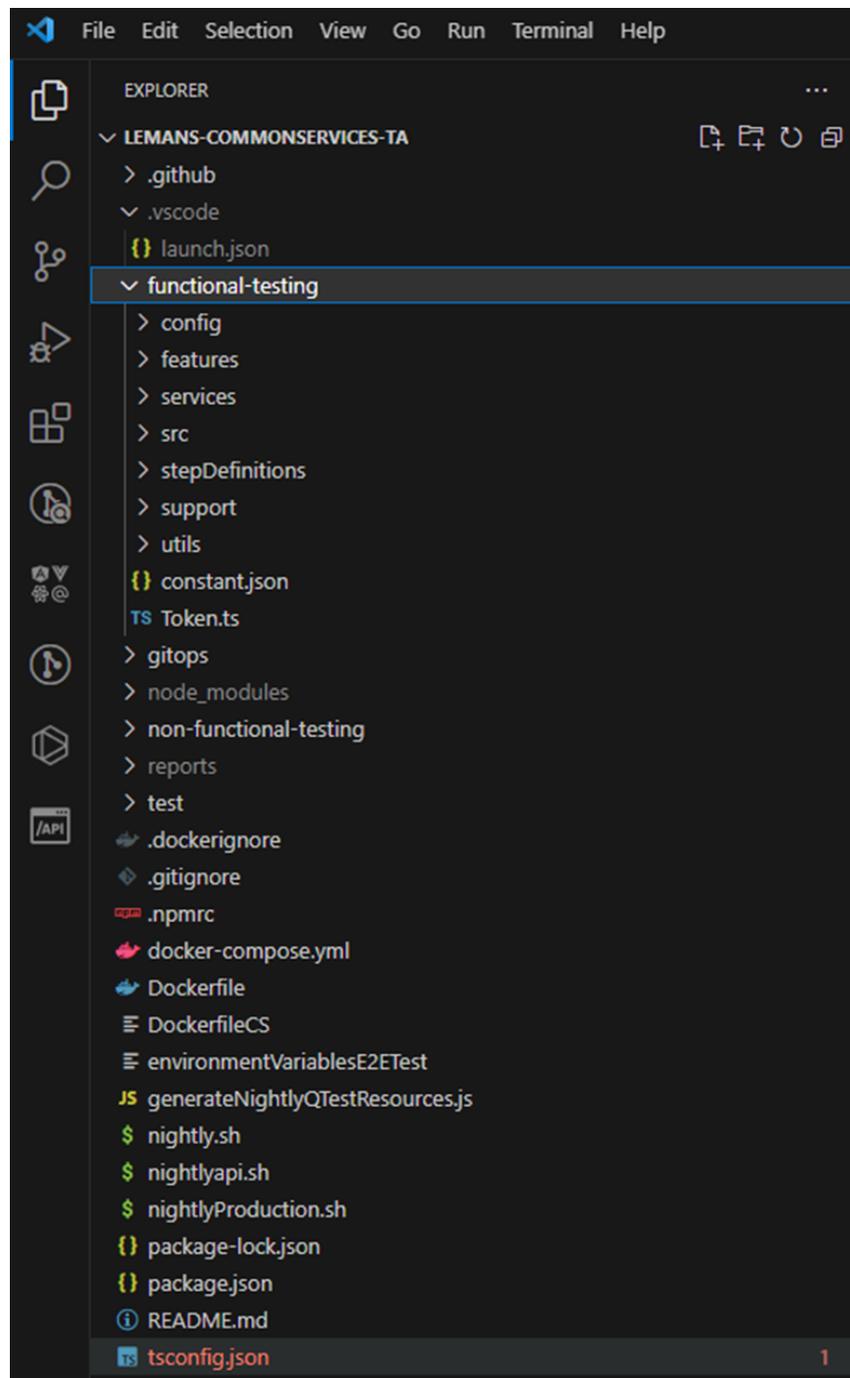


Figura 9. Estructura del código del proyecto.

El repositorio cuenta con diversas carpetas con el fin de tener una organización interna y así facilitar el trabajo sobre el mismo. La carpeta de functional-testing cuenta con todo lo necesario para administrar las pruebas funcionales, cuyo objetivo es verificar el sistema, es decir, que este funcione correctamente. Por otra parte, la carpeta non-functional-testing se enfoca en las

pruebas no funcionales, cuyo objetivo es medir qué tan bien el sistema cumple con las tareas requeridas. Allí se encuentran scripts y configuraciones para lanzar pruebas de rendimiento sobre el aplicativo.

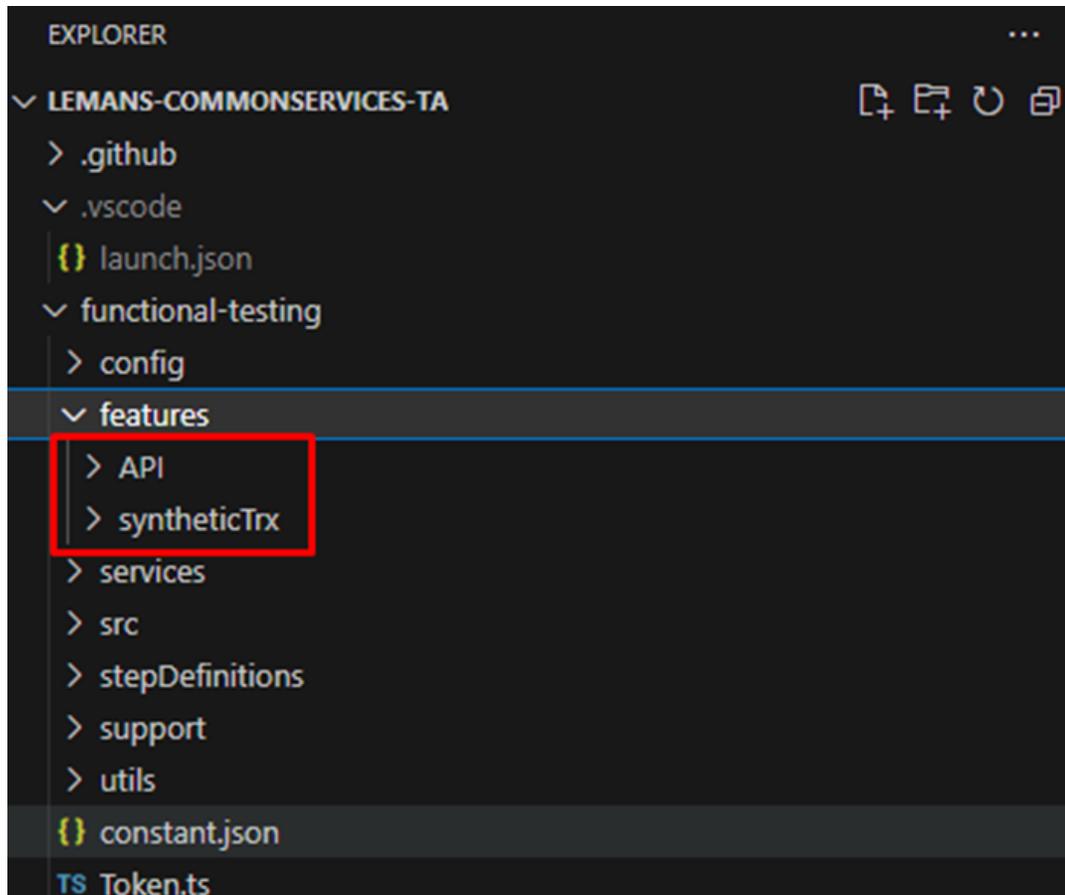


Figura 10. Estructura interna de las pruebas funcionales

Las pruebas funcionales se enfocan no solo por interfaz gráfica (GUI), sino también por servicio o API. Por tal motivo, se diferencian dentro del proyecto usando una carpeta para cada tipo de prueba respectivamente.

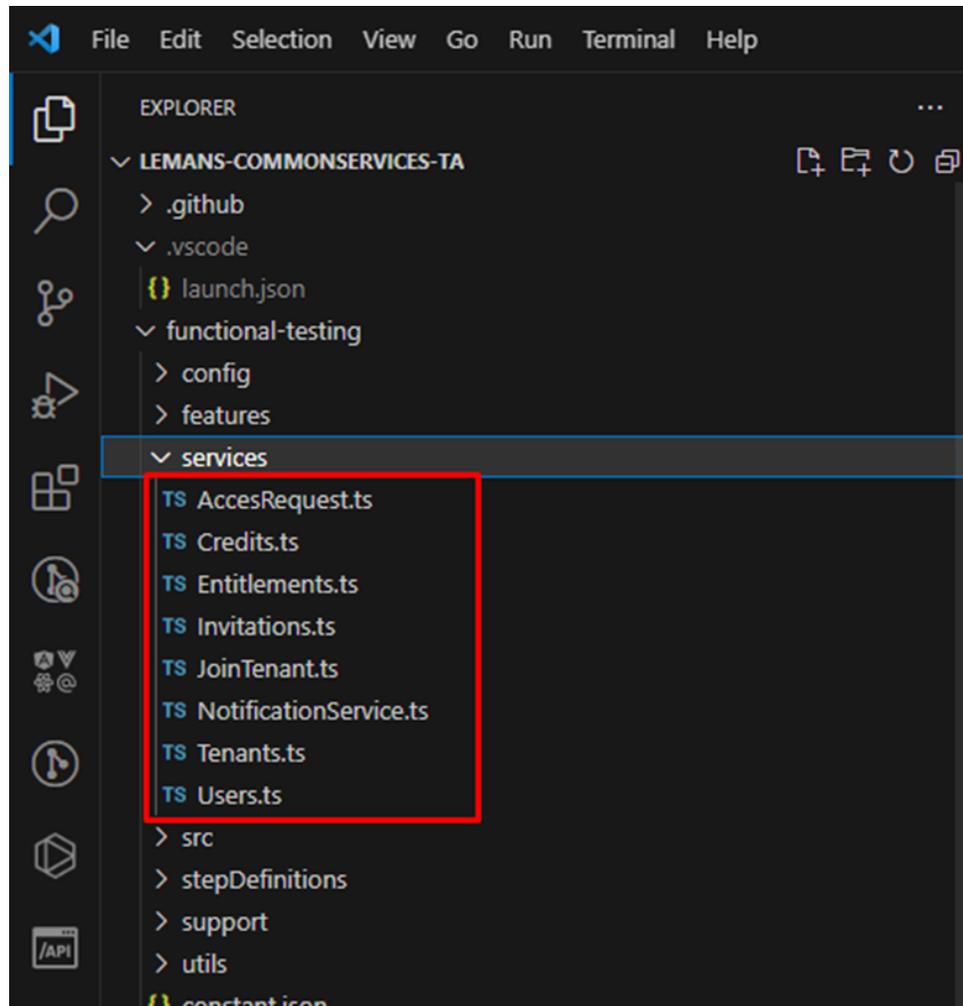


Figura 11. Archivos para la ejecución de pruebas

En la carpeta de servicios se encuentran los archivos TypeScript cuyo contenido se basa en las operaciones (funciones) que se llaman y usan desde diversas partes del código.

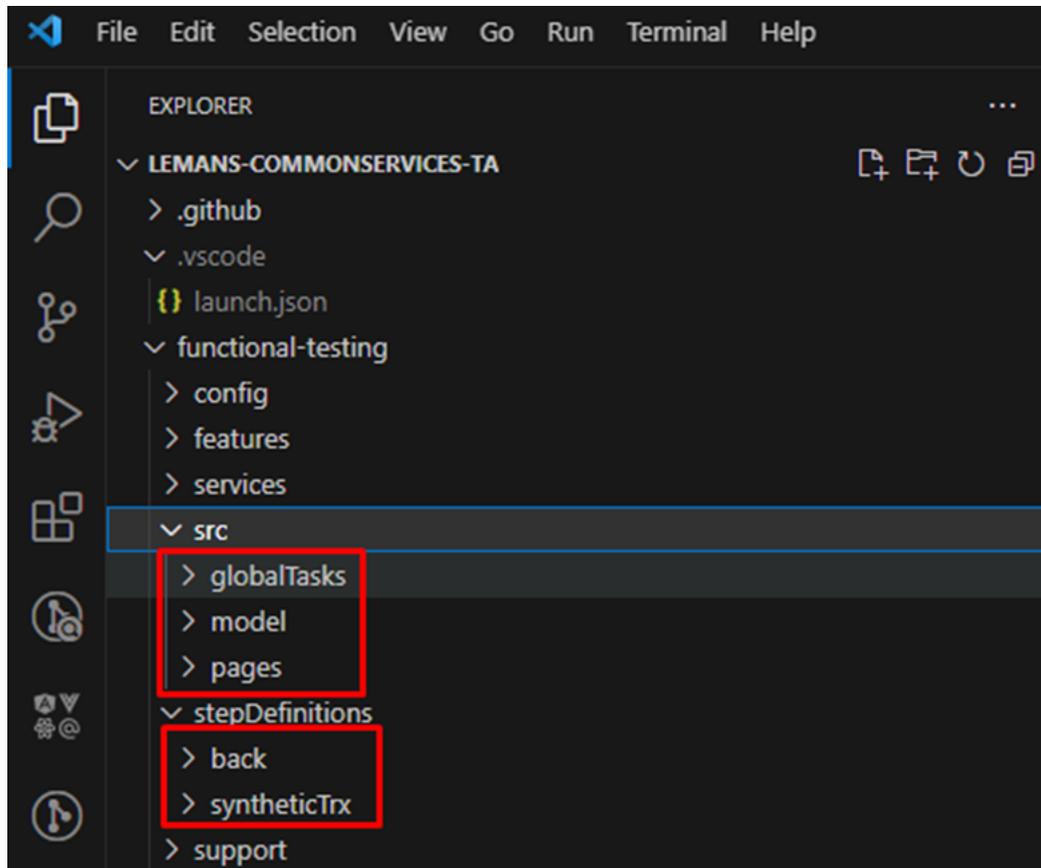


Figura 12. Carpeta src

La carpeta src es la más importante al interior del proyecto, pues allí se aloja todo el código y scripts de las pruebas automatizadas, mediante los cuales, estas funcionan, se actualizan o incluso se eliminan.

## V. RESULTADOS

Durante el periodo de prácticas se trabajó siempre en la búsqueda de mejorar el proceso de pruebas dentro de la compañía y lo que al mismo respecta, el progreso a nivel de automatización fue significativo, ya que como se muestra en la siguiente tabla, el número total de escenarios de prueba automatizados en el último (Sprint 49) respecto al primero (Sprint 36) fue de más del doble de casos nuevos automatizados.

TABLA I  
MÉTRICAS DE AUTOMATIZACIÓN POR SPRINT

Sprint	Total de escenarios automatizados	Pruebas API	Pruebas UI	Anomalías	Cobertura total de automatización de pruebas
36	124	27	97	3	92.5%
37	132	27	105	13	92.9%
38	139	30	109	10	93.2%
39	139	30	109	11	93.2%
40	157	42	115	10	93.4%
41	160	46	116	8	93.5%
42	201	81	120	10	98.4%
43	222	101	121	4	99.1%
44	232	111	121	9	98.7%
45	237	115	122	6	98.7%
46	237	115	122	6	92.9%
47	242	115	127	8	94.9%
48	245	115	130	9	96.7%
49	264	115	149	7	98.8%

No solo se puede apreciar el buen incremento en el total de escenarios automatizados con respecto al sprint inicial, sino también, el hecho de que la cobertura total de automatización de pruebas tuvo una tendencia casi que en todos los casos a incrementar como se muestra a continuación, esto lo que significa es que a medida que el proyecto crecía y nuevas funcionalidades iban siendo incluidas y desplegadas en el aplicativo, el proceso de pruebas se realizaba a una velocidad y eficiencia tal que los casos por cubrir o casos de prueba faltantes en cada sprint eran muy pocos.

Evolución: nº casos automatizados vs nº casos por cubrir

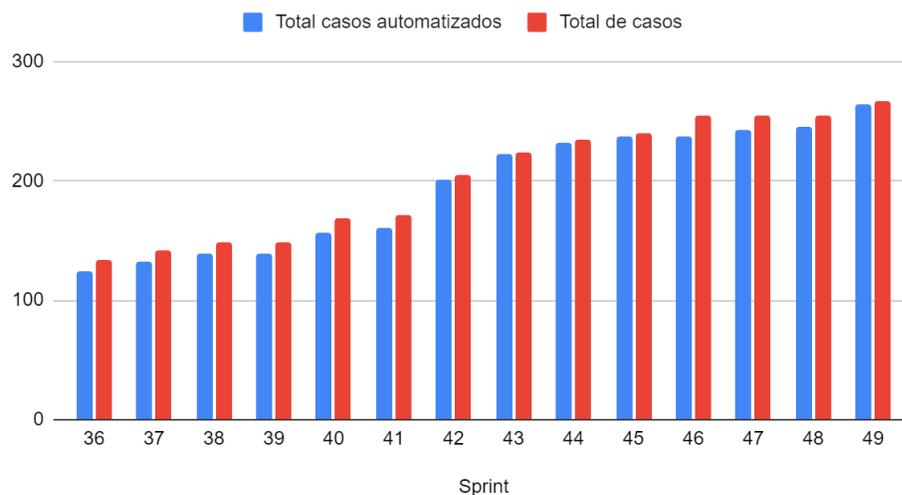


Figura 13. Evolución total casos automatizados vs casos por cubrir

Adicionalmente, se mantuvo constante diseño y mantenimiento sobre los scripts automatizados para pruebas tanto por interfaz como por servicio, haciendo que el número de escenarios totales en ambos casos incrementará a lo largo del tiempo, tal y como se aprecia en la tabla I. Por otra parte, se reportaron anomalías en cada uno de los sprints (ver figura 14), sin embargo, por confidencialidad de la compañía no es posible dar detalles más específicos de los mismos. Aun así, se siguió el debido proceso respecto al tratamiento de las mismas realizando pruebas para verificar que hayan sido solventados correctamente, tal y como se mencionó en la sección de metodología.

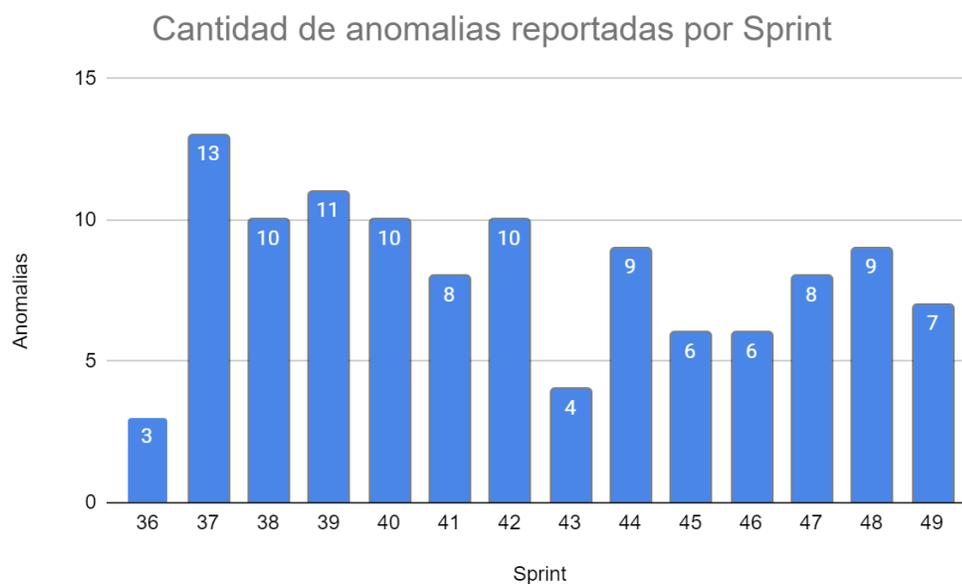


Figura 14. Cantidad de anomalías reportadas por Sprint

Finalmente es importante mencionar que al inicio, no se hacía un uso tan extenso de la herramienta QTest, desaprovechando todo lo que la misma ofrece a lo que respecta un proceso de pruebas, sin embargo, con el paso del tiempo y atendiendo las sugerencias de los stakeholders, líderes del proyecto e incluso otros equipos, se empezó a hacer un uso verdaderamente importante para el desarrollo de las pruebas en el proyecto, apoyando los procesos de pruebas de regresión, de despliegues y generando información valiosa en cada iteración.

## VI. CONCLUSIONES

Finalmente se lograron los objetivos planteados e incluso obteniendo mejores resultados de los esperados, ya que se cubrieron partes internas del proceso de pruebas dentro del proyecto que al inicio no se habían planteado.

- Cada despliegue es acompañado por diversos conjuntos de pruebas totalmente automatizadas que se ejecutan en la nube, scripts totalmente mantenibles y escalables que permiten rectificar en todo momento el estado de las diversas funcionalidades del aplicativo FactoryTalk Hub, permitiendo así, asegurar la calidad del mismo.
- Las pruebas automatizadas no solo se enfocan en verificar la interfaz gráfica, sino también la calidad de los servicios, APIs e integración de los diversos componentes que se comunican constantemente en el funcionamiento de la aplicación.
- Durante cada sprint se detectaron anomalías las cuales seguían un proceso de corrección por parte del equipo de desarrollo y posteriormente una validación mediante un test de confirmación para así asegurar que la misma no se siguiera replicando en la aplicación. Además, de que se documentaba este proceso para tener una trazabilidad clara sobre los errores presentados en caso de una réplica posteriormente.
- Las métricas de automatización se mejoraron, tal y como se evidencia en la sección de resultados, el número de casos automatizados y revisados por sprint creció a una media de 6.2% de sprint a sprint y el porcentaje de cobertura se mantuvo en números mínimo del 92,5%, y una media del 95,5%.

## REFERENCIAS

- [1] BOSE, Shreya. UI Testing: A Detailed Guide | BrowserStack. BrowserStack [página web]. (22, diciembre, 2022). [Consultado el 18 de marzo de 2023]. Disponible en Internet: <<https://bit.ly/47CWiUb>>.
- [2] FERNÁNDEZ, Yúbal. API: qué es y para qué sirve. Xataka - Tecnología y gadgets, móviles, informática, electrónica [página web]. (23, agosto, 2019). [Consultado el 20, abril, 2023]. Disponible en Internet: <<https://bit.ly/3Yz82CZ>>.
- [3] GILLIS, Alexander S. y BRUSH, Kate. What Is API Testing? | Definition from TechTarget. App Architecture [página web]. (7, marzo, 2023). [Consultado el 20, abril, 2023]. Disponible en Internet: <<https://bit.ly/3YAQyWQ>>.
- [4] BOSE, Shreya. Difference between Bugs, Errors and Issues | BrowserStack. BrowserStack [página web]. (30, agosto, 2021). [Consultado el 22, abril, 2023]. Disponible en Internet: <<https://bit.ly/47ul87z>>.
- [5] BOSE. How to write Test Cases (with Format & Example) | BrowserStack. BrowserStack [página web]. (6, febrero, 2023). [Consultado el 22, abril, 2023]. Disponible en Internet: <<https://bit.ly/3OBGmIV>>.
- [6] BITBUCKET. Qué es el control de versiones | Atlassian Git Tutorial. Atlassian [página web]. (16, octubre, 2018). [Consultado el 22, abril, 2023]. Disponible en Internet: <<https://bit.ly/45yaOui>>.
- [7] P. Modispacher. "What is Jira, and why use it?" Actonic – Unfolding your potential. <<https://bit.ly/3OYcsjx>> (accedido el 16 de mayo de 2023).
- [8] Lutkevich, B. (2021, octubre 28). Scrum. Software Quality; TechTarget. <<https://bit.ly/456oe0S>>
- [9] ¿Qué es la prueba de software y cómo funciona? | IBM. (2017, 30 de octubre). IBM - Deutschland | IBM.<<https://ibm.co/47ZrEEG>>.

## VII. ANEXOS

### Glosario de términos:

- **Pruebas por interfaz (UI):** Las pruebas por interfaz de usuario, o también conocidas como GUI, son básicamente un mecanismo para probar los elementos visuales de un aplicativo web o móvil con el fin de verificar que están funcionando acorde a los requisitos, en términos de funcionalidad y desempeño, estas pruebas aseguran que la interfaz de usuario (UI) esté libre de anomalías. [1].
- **API:** La interfaz de programación de aplicaciones (API), es un conjunto de definiciones y protocolos que permite la comunicación entre dos aplicaciones o módulos de software para cumplir una o muchas funciones. [2]
- **Pruebas por servicio (API):** En este tipo de pruebas se analiza una API, con el fin de verificar que esta cumple con la funcionalidad, seguridad, desempeño y fiabilidad esperada. Las pruebas se realizan directamente en la API o como parte de una prueba de integración.[3]
- **Anomalía:** Se refiere a la presencia de un resultado diferente al esperado, no necesariamente es un error del aplicativo, también puede relacionarse con problemas de usabilidad. Es tarea del Test Automation Engineer o tester, definir cuál de ambas es. [4]
- **Escenario de prueba:** Son acciones requeridas para verificar una funcionalidad específica durante las pruebas de software. Un caso de prueba detalla los pasos, datos, prerrequisitos y post condiciones necesarias para verificar una característica o funcionalidad. [5]
- **Sprint / interacción:** Un sprint, es un concepto del agilismo que hace referencia a un periodo de tiempo determinado para el cual se definen entregables y objetivos, los cuales deben ser completados antes de la fecha de finalización de este.
- **Métricas de automatización:** Son datos relevantes que surgen del proceso de automatización de pruebas de software, tal y como el número de escenarios de pruebas

automatizados que se tienen, el porcentaje de casos fallidos/exitosos y la cantidad de anomalías o bugs presentes en dichos escenarios.

- **Control de versiones:** También conocido como control de código fuente, es la práctica de rastrear y gestionar los cambios en el código de software. Los sistemas de control de versiones son herramientas de software que ayudan a los equipos de software a gestionar los cambios en el código fuente a lo largo del tiempo [6]. Rockwell Automation Company 'Public'
- **Jira:** Es una aplicación de software desarrollada por la empresa de software australiana Atlassian que permite a los equipos realizar un seguimiento de los problemas, gestionar proyectos y automatizar los flujos de trabajo. [7]
- **SCRUM:** Scrum es un marco para la gestión de proyectos que enfatiza el trabajo en equipo, la responsabilidad y el progreso iterativo hacia un objetivo bien definido. El marco comienza con una premisa simple: comience con lo que se puede ver o conocer. Después de eso, realice un seguimiento del progreso y ajuste, según sea necesario. [8]
- **Créditos:** Son recursos virtuales que son adquiridos por los clientes de la compañía a través de dinero real, mediante los cuales, dichos clientes pueden adquirir, provisionar y finalmente utilizar los servicios que requieran.
- **Entitlements:** Son contratos comprados a través de créditos, mediante los cuales un usuario/cliente tiene derecho a un determinado servicio por un periodo de tiempo determinado.
- **Provisionar:** Es el proceso mediante el cual, el usuario/cliente decide qué cantidad de créditos le asigna a un determinado servicio, tomando en cuenta que, entre mayor sea el número de créditos que se asignen, mayor tiempo podrá hacer uso del mismo.
- **Servicio:** Son diferentes aplicativos de los cuales el usuario/cliente puede hacer uso, tal y como los mencionados anteriormente: : Arena™, Optix™, Studio 5000™, Remote Access™, Plex™ y Fix™. Cada uno de estos, ofrece diferentes características de acuerdo a las necesidades del cliente u organización.