



**Automatización de Pruebas en People App: Incrementando la Eficiencia y la Precisión en el
Proceso de Pruebas**

Sebastian Ochoa Gonzalez

Informe final de la práctica empresarial para optar al título de Ingeniero de Sistemas otorgado por
la Universidad de Antioquia

Asesor

Andrés David Santamaría Galvis, Ph.D.

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería de Sistemas

Medellín

2024

Cita	(Sebastian Ochoa Gonzalez, 2024)
Referencia	Ochoa Gonzalez, S. (2024). <i>Automatización de Pruebas en People App: Incrementando la Eficiencia y la Precisión en el Proceso de Pruebas</i> [Práctica empresarial]. Universidad de Antioquia, Medellín.
Estilo APA 7 (2020)	



Centro de Documentación Ingeniería (CENDOI)

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

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Julio César Saldarriaga Molina.

Jefe departamento: Diego José 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.

Dedicatoria

A mi madre Alcira, que con todo el sudor de su esfuerzo me ha criado y educado para ser quien soy ahora. No estaría aquí alcanzando mis metas de no ser por ella.

Agradecimientos

Mis agradecimientos se extienden a GAP por haberme dejado culminar mi carrera universitaria con normalidad, especialmente a Ana Chevez por todo su apoyo técnico y humano al interior de la organización. También a Andrés Santamaría por su disposición y colaboración en el correcto desarrollo de este trabajo.

Tabla de contenido

Resumen	8
Abstract	9
Introducción	10
1 Objetivos	12
1.1 Objetivo general	12
1.2 Objetivos específicos	12
2 Marco teórico	13
2.1 Casos de prueba	13
2.2 Automatización de pruebas	13
2.3 Mantenimiento de pruebas automatizadas	14
2.4 ¿Cómo se van a realizar estas actividades durante el desarrollo del proyecto?	15
2.4.1 Metodologías ágiles	15
2.4.2 Scrumban	16
2.5 ¿Qué beneficios traería la automatización de pruebas en People App?	16
3 Metodología	17
4 Resultados	21
5 Análisis	25
6 Conclusiones	26
7 Recomendaciones	27
Referencias	28
Anexos	28
Anexo 1. Propuesta del proyecto de práctica	28

Lista de tablas

Tabla 1 Cantidad de casos de prueba de los módulos terminados	21
Tabla 2 Cantidad de casos de prueba de los módulos no terminados	22

Lista de figuras

Figura 1 Ciclo de semanas del equipo de automatización	18
Figura 2 Primera parte del resultado de la ejecución del suite de pruebas	23
Figura 3 Segunda parte del resultado de la ejecución del suite de pruebas	23

Siglas, acrónimos y abreviaturas

GAP	Growth Acceleration Partners
Auto	Automatización
Mnto	Mantenimiento

Resumen

El proyecto People App es una iniciativa de *software* interna de la compañía GAP, para facilitar y documentar los procesos administrativos más comunes que se realizan dentro de la organización, como lo son: la asignación de personal, la asignación de clientes, el seguimiento motivacional de los empleados, el seguimiento de los logros personales y las certificaciones adquiridas, entre otras actividades.

Actualmente, el aplicativo People App cuenta con una gran cantidad de módulos, y por lo tanto un número significativamente alto de casos de prueba que consumen muchísimo tiempo en pruebas manuales, sobre todo en la ejecución de pruebas de regresión, por lo que se implementó un desarrollo de pruebas automatizadas que agilizó enormemente este proceso facilitando la entrega continua, esencial para la metodología ágil sobre la cual se está desarrollando este proyecto.

Este proyecto resultó en un aumento significativo del número de pruebas automatizadas logrando que el proceso de la evaluación del *software* se ejecute de forma mucho más rápida, disminuyendo el tiempo entre cada entrega, aumentando la precisión de las pruebas y su confiabilidad. Simultáneamente se realizó un mantenimiento constante al suite de pruebas automatizadas, manteniéndolas actualizadas ante los cambios y asegurando un alto nivel de estabilidad en su ejecución.

Teniendo en cuenta el esfuerzo realizado en el desarrollo de este proyecto y los resultados obtenidos, se pueden evidenciar los beneficios que trae la automatización de pruebas dentro del desarrollo de *software*, siempre y cuando se tengan en cuenta sus limitaciones.

Palabras clave: pruebas de *software*, automatización de pruebas, casos de prueba, entrega continua, mantenimiento de pruebas.

Abstract

The People App project is an internal software initiative of the GAP company to document and facilitate the most common administrative processes carried out within the organization, such as personnel assignment, customer assignment, monitoring employee motivation, keeping track of personal achievements, and new acquired certifications, among other activities.

Currently, the People App application has a large number of modules and therefore a significantly high number of test cases that consume a lot of time in manual testing, especially in the execution of regression tests, which is why automated tests were developed, greatly streamlining this process and facilitating continuous delivery, which is essential for the agile methodology on which this project is being developed.

This project resulted in a significant increase in the number of automated tests, making the software evaluation process run much faster, reducing the time between each delivery, and increasing the precision of the tests and their reliability. Simultaneously, constant maintenance was carried out on the automated test suite, keeping them updated with new changes and ensuring a high level of stability in their execution.

Considering the effort made in the development of this project and the obtained results, the benefits that test automation brings to software development can be evident, as long as its limitations are taken into account.

Keywords: software tests, test automation, test cases, continuous delivery, test maintenance.

Introducción

Growth Acceleration Partners, en adelante GAP, es una compañía que brinda soluciones tecnológicas a través del desarrollo personalizado de *software* y servicios de consultoría a distintos tipos de empresas, por lo que requiere una administración precisa y efectiva de sus recursos humanos y tecnológicos. Actualmente GAP hace uso de plataformas web de terceros para estas tareas, por lo que People App, es de manera resumida, la unión de estas plataformas en una única aplicación web propia de la compañía. Este aplicativo cuenta actualmente con un alto número de módulos y, en consecuencia, una gran cantidad de casos de prueba que suelen ser ejecutados manualmente por un promedio de una a dos personas, situación que resulta en un extenso tiempo de espera entre cada entrega del aplicativo dada la larga duración de la ejecución manual de las pruebas.

Por consiguiente, se decidió realizar la implementación de pruebas automatizadas para agilizar el proceso de evaluación del aplicativo y realizar entregas mucho más constantes, además de entre otros beneficios que conlleva la automatización de pruebas como lo pueden ser la confiabilidad y la precisión.

Así pues, al iniciar el desarrollo ya se tenía una cantidad reducida de pruebas automatizadas que hasta el día de hoy se corren diariamente con el fin de mantener una visión constante del estado de la aplicación, sin embargo no ofrecían un ahorro significativo en el tiempo que tomaba realizar la evaluación del *software* de forma manual, y aunque actualmente este tiempo ha mejorado indudablemente, sigue siendo considerablemente alto, por lo que este proyecto continuará en desarrollo de forma indefinida.

Ahora bien, dado que People App es una aplicación realizada bajo el marco de una metodología ágil basada en Scrumban, es necesario considerar que ésta puede cambiar sus requerimientos en cualquier momento y por tanto se deben mantener actualizados los casos de prueba que hayan sido afectados por estos cambios y dar un mantenimiento constante a las pruebas automatizadas que puedan estar fallando, sea por su propia inestabilidad o por los nuevos requerimientos efectuados.

Por lo anterior, podemos entonces concluir que, además del objetivo de aumentar el número de pruebas automatizadas, este proyecto también debe asegurar su estabilidad en el

tiempo y su actualización mediante un mantenimiento constante, sin dejar de lado el aumento persistente en sus números.

En síntesis, este texto tiene como finalidad documentar el proceso y el resultado de la implementación de la automatización de pruebas y su constante mantenimiento en el proyecto People App con el objetivo de reducir los tiempos de ejecución de la evaluación del aplicativo, cuyo alcance en un inicio se planteó en 4 módulos automatizados en su totalidad, pero que se logró superar, resultando en un total de 7 módulos, además de otra gran cantidad de pruebas repartidas en los 8 módulos restantes, aunque teniendo como principal enfoque los módulos completados.

1 Objetivos

1.1 Objetivo general

Disminuir el tiempo invertido en la ejecución manual del proceso de pruebas para así agilizar las entregas funcionales de la aplicación.

1.2 Objetivos específicos

- Analizar los casos de prueba para definir cuales son automatizables y agregarlos a una lista de tareas.
- Aumentar la cantidad de pruebas automatizadas para disminuir el esfuerzo de la ejecución de pruebas manuales.
- Realizar un mantenimiento constante a las pruebas automatizadas para mantenerlas estables y actualizadas a través del tiempo.

2 Marco teórico

El interés de este proyecto son concretamente los largos tiempos de espera en la ejecución de pruebas de regresión antes de cada entrega. Pero ¿qué son las pruebas de regresión? Este es uno de los varios conceptos que se deben tener claros para la realización de este proyecto y que vamos a explicar a continuación:

2.1 Casos de prueba

Al momento de probar un aplicativo de *software*, es necesario tener en cuenta cuáles son los requisitos para definir si éste cumple o no con las necesidades del agente interesado. Los casos de prueba no son más que un insumo de precondiciones, entradas, acciones y resultados esperados (ISTQB, 2023) con los que validar que el *software* toma en cuenta estos requisitos y se comporta de forma correcta.

Existen varios tipos de pruebas a partir de los cuales podemos crear casos de prueba que cumplan con las distintas validaciones requeridas. Dentro del proyecto People App el tipo de pruebas al que se le dará una mayor prioridad son las pruebas de regresión, estas se ejecutan luego de que se haya realizado alguna modificación a la aplicación y tienen como fin detectar posibles defectos que hayan podido introducirse en sectores que no fueron modificados y que ya hayan sido probados previamente (ISTQB, 2023), lo que en una aplicación con tantos módulos y actividades como lo es People App, se traduce en una gran cantidad de tareas repetitivas en pruebas y un largo tiempo de ejecución de las mismas, a partir de esta problemática es que se deciden implementar la automatización, pero ¿Que es la automatización de pruebas?

2.2 Automatización de pruebas

Dada la gran cantidad de actividades repetitivas que se realizan en un proceso de pruebas de un proyecto de *software*, se vió la necesidad y surgió la posibilidad de hacer estos procesos de forma automática, ahorrando esfuerzo y tiempo de los equipos de desarrollo. Así surgen entonces tecnologías y *frameworks* de desarrollo de *software* que lo hacen posible, como lo son: Selenium, Cypress, Playwright entre otras tecnologías. Así, quien automatiza estas pruebas debe en pocas

palabras reescribir los casos de prueba en código, con sus precondiciones, acciones y resultados esperados, que luego son interpretados, ejecutados y validados por las herramientas mencionadas.

Para el proyecto People App se decidió usar el *framework* Playwright ya que posee una ventaja muy específica que es de máxima importancia para el uso de la aplicación, específicamente en las pruebas automatizadas: Única autenticación (Playwright, 2023). Dado que, para el uso correcto del aplicativo en cuestión es imprescindible autenticarse, esta ventaja ayuda a que la ejecución sea más rápida evitando que se realice un inicio de sesión antes de cada prueba pues esta acción solo la realiza una única vez al inicio y usa esta misma sesión para ejecutar las pruebas, esto también evita que el robot pueda ser tomado como un ataque a la aplicación dados los múltiples inicios de sesión que podrían generarse de una herramienta que no posea esta cualidad.

Una vez creadas las pruebas automatizadas, es posible que con el tiempo empiecen a dar resultados falsos positivos o falsos negativos, por lo que éstas requieren un mantenimiento constante.

2.3 Mantenimiento de pruebas automatizadas

Mantenimiento es la acción de modificar un producto después de terminado, para corregir defectos, mejorar su calidad y adaptarlo a los cambios (ISTQB, 2023). Aplicando este término a las pruebas automatizadas, podríamos tomar el producto terminado como las pruebas que ya han sido añadidas, pero que por algún motivo estén fallando, sea por alguna actualización del aplicativo o algún tema de inestabilidad de la forma en la fue escrita la prueba.

Sin embargo, el mantenimiento no puede ser un término aislado, pues ¿Cómo nos damos cuenta de cuáles pruebas están fallando y necesitan ser corregidas? Para identificar cuáles tests están fallando haremos uso de un pipeline de Gitlab. Un pipeline es un componente que ejecuta ciertas tareas en un determinado orden, al cual se le puede dar una multitud de configuraciones para que se comporte de una manera que cumpla con las necesidades del proyecto (GitLab, 2023).

Aunque este componente suele utilizarse en conjunto con otros para realizar entregas continuas de la aplicación, esta vez estará configurado para ejecutar únicamente el suite de pruebas automatizadas cada 24 horas, y sin importar el resultado de la ejecución (sea positivo o

negativo) desplegará los resultados en una página web desde la cual se podrán consultar. Gracias a la herramienta Playwright estos resultados arrojan un conjunto de datos relacionados a cada prueba como lo son los pasos ejecutados, screenshots y videos, de este modo podemos consultar cada uno de los test fallidos e identificar si el causante es un error de la aplicación, un cambio realizado a la página o un tema de inestabilidad en el código de la prueba, lo que finalmente nos permitirá realizar un mantenimiento adecuado al suite de pruebas automatizadas.

2.4 ¿Cómo se van a realizar estas actividades durante el desarrollo del proyecto?

Para esto es necesario que definamos qué son las metodologías ágiles y qué es Scrumban, siendo esta última la metodología seleccionada por ser una de las más apropiadas para proyectos grandes sin fecha límite cómo es el caso de People App.

2.4.1 Metodologías ágiles

En el ámbito del desarrollo de *software* existen varias etapas que de forma resumida son: la toma de requerimientos, el diseño, la construcción, la verificación y finalmente la entrega. En las metodologías de trabajo tradicionales se hacía cada una de estas etapas una única vez a lo largo del proyecto, mientras que en las metodologías ágiles se realizan estas etapas de forma cíclica en cortos periodos de tiempo de modo que las entregas sean constantes y además de ellas se pueda recibir retroalimentación inmediata (Sarah Laoyan, 2022). Este tipo de metodologías ofrece varias ventajas respecto a las tradicionales, el mayor ejemplo es que gracias a la retroalimentación inmediata, un proyecto que trabaja bajo el marco de una metodología ágil siempre está dispuesto al cambio y a la adaptación sin interrumpir el flujo de trabajo (Sarah Laoyan, 2022), lo cual es determinante para la realización de este proyecto, por esto, es importante realizar un constante mantenimiento al suite de pruebas automatizadas, para mantenerlo actualizado frente a estos posibles cambios.

2.4.2 Scrumban

Como lo define Sarah Laoyan (2022), Scrumban es una metodología ágil de gestión de proyectos que combina las mejores características de dos de las metodologías más conocidas: Scrum y Kanban. De Scrum posee reuniones de retrospectiva en las cuales se identifica qué factores pueden ser mejorables para que el equipo de desarrollo mantenga un buen ritmo de trabajo de manera saludable, abarcando desde lo técnico como puede ser mejorar el uso de alguna herramienta, hasta lo más humano como mejorar la comunicación entre el equipo. De Scrum también posee la revisión del resultado de cada ciclo, donde se presenta al interesado el estado actual de la aplicación con las actualizaciones nuevas que se incluyeron durante el ciclo de trabajo. Así mismo, también tiene la priorización del trabajo y la definición de lo que significa que una tarea haya sido finalizada de la metodología Scrum, lo cual permitirá dar un orden y un estado de finalización a las tareas pendientes por realizar.

Simultáneamente, Scrumban posee de Kanban la utilización de tableros de actividades donde se podrá visualizar de manera más simple y gráfica el estado de las tareas pendientes, las que están en desarrollo y las terminadas. De Kanban también hereda la libertad del equipo de administrar sus tareas y la limitación de tareas por persona para evitar el sobretrabajo (Sarah Laoyan, 2022).

A pesar de la definición de actividades que nos dan las metodologías ágiles, lo normal es que los equipos de desarrollo no se adhieran estrictamente a seguirlas paso por paso, es por esto que la mayoría de ellos no las consideran metodologías, sino más bien marcos de trabajo, pues en realidad no son más que una base para crear una metodología propia y sólida donde se definirán las actividades que más se adaptan a las necesidades del proyecto y que permitan tener un ritmo constante y saludable de trabajo y en este caso no es una excepción, ya que para el desarrollo de este proyecto tomaremos solo algunas de estas actividades cómo se detallarán más adelante en la metodología del proyecto.

2.5 ¿Qué beneficios traería la automatización de pruebas en People App?

Automatizar los procesos repetitivos de pruebas tiene una gran cantidad de beneficios. Además de ahorrar tiempo, esta práctica también reduce el margen de error humano, aumenta la

eficiencia de los equipos, provee una increíble facilidad en el seguimiento de las métricas, mejora la calidad entre múltiples plataformas y reduce costos a largo plazo, lo que finalmente se traduce en entregas más rápidas sin sacrificar la calidad del *software*, al contrario con un grado de calidad más alto (Jennifer Castro, 2021). Esto en People App no será distinto, pues están alineados con el objetivo principal de reducir el tiempo de ejecución manual de pruebas, para así realizar entregas más constantes y de mejor calidad.

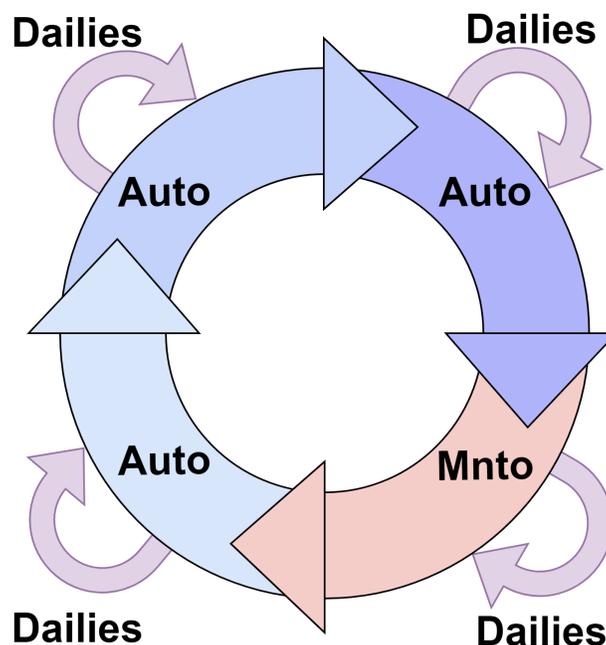
3 Metodología

El marco metodológico sobre el cual se trabajó para la realización de este proyecto fue Scrumban, ahora bien, el equipo de automatización de pruebas trabajó de manera independiente al equipo de desarrollo de la aplicación. Así, el equipo de automatización tuvo su propia versión de la metodología y es en la cual nos centraremos en este documento puesto que es donde se realizó el foco del proyecto, que fue la automatización de pruebas.

El equipo de automatización no guió su avance en periodos de tiempo de dos semanas como es común en equipos que trabajan bajo Scrumban, más bien, cada integrante tuvo su propio ciclo compuesto por semanas de automatización y una semana de mantenimiento, de modo que ésta última se turnó entre cada miembro del equipo y así se mantuvo un mantenimiento constante al suite de pruebas. Teniendo en cuenta que el equipo estaba compuesto por cuatro individuos, se tuvo un ciclo indefinido de cuatro semanas, donde tres de estas fueron dedicadas completamente a la automatización y la última al mantenimiento, como podemos observar en la siguiente ilustración (Figura 1):

Figura 1

Ciclo de semanas del equipo de automatización



Cada semana de automatización tuvo como objetivo desarrollar un caso de prueba por día, pero teniendo en cuenta que este es un escenario ideal, se asumieron posibles excepciones en las que una prueba pueda haber tomado alrededor de 2 días y así se consiguió un promedio semanal de 4 pruebas automatizadas a la semana por cada integrante del equipo de automatización, sin tener en cuenta los días feriados.

Mientras que cada semana de mantenimiento tuvo los siguientes objetivos que se desarrollaron según fuese necesario:

- Reportar errores de la aplicación encontrados en la ejecución de las pruebas.
- Corregir las pruebas que están fallando por actualizaciones de la aplicación.
- Estabilizar las pruebas que están fallando por inconsistencias en el código.
- Corregir elementos de forma dentro del código (según las convenciones establecidas por el equipo; como escritura de métodos o variables, entre otros pequeños detalles).

Siendo el más importante la corrección y estabilización de pruebas, con la meta de mantener la tasa de fallos en un porcentaje menor al 10 % en el reporte diario resultado de la ejecución del suite de pruebas automatizadas. Esta ejecución se realizó en un pipeline de GitLab todos los días de la semana a media noche, con el fin de mantener constante visibilidad sobre el estado de la aplicación.

Al mismo tiempo que se realizaban las actividades descritas anteriormente, también se llevaban a cabo algunas de las reuniones clásicas de Scrumban, como lo son los dailies y las retrospectivas. En primer lugar, los dailies fueron reuniones diarias en las cuales se compartía con el equipo el estado de las tareas de las que se era responsable, indicando lo que se hizo el día anterior y lo que se haría a continuación, además de si se encontró algún inconveniente que requiera la atención del equipo o que pudiera retrasar la terminación de estas tareas. Mientras tanto, las retrospectivas se realizaron una vez al mes con el fin de identificar mejoras en el flujo de trabajo del equipo, ya fuese por el uso de las herramientas tecnológicas o la forma de trabajo de los integrantes, o también, para destacar el buen trabajo de alguno de los miembros del equipo.

Ahora bien, el manejo de las tareas se realizó a través de tableros en la aplicación Trello, donde se distribuyeron en 3 columnas de la siguiente manera:

- **Backlog:** Donde existía una tarea por cada caso de prueba pendiente de automatizar cuya viabilidad ya había sido previamente verificada.
- **Doing:** Esta columna era para los casos de prueba (tareas) que se estaban automatizando en el momento.
- **Code Review:** Aquí estuvieron los casos de prueba automatizados cuyos scripts estaban listos para la revisión por parte de otro integrante del equipo que validaría el correcto uso de convenciones y buenas prácticas en el código fuente.
- **Done:** Columna donde se estarían anexado los casos de prueba (tareas) que ya habían sido automatizados, revisados y agregados al suite de pruebas principal.

Cada tarea tenía un id, un nombre, una descripción, un miembro del equipo responsable y un puntaje asignado que representaba la complejidad de la tarea basado en la escala fibonacci, donde el más simple era 1 y a medida que aumentaba la complejidad también subía este número.

Así pues, cada integrante del equipo tenía acceso a una hoja de cálculo donde estaban almacenados los casos de prueba escritos anteriormente por el equipo de pruebas manuales y era libre de agregar más tareas a la columna backlog basándose en este archivo y/o asignarse una tarea que ya estuviese allí previamente, de modo que cada miembro era responsable de administrar sus propias actividades.

En cuanto a las herramientas utilizadas para el desarrollo de este proyecto están como *framework* de automatización Playwright, como lenguaje de programación JavaScript, como repositorio y encargado de las ejecuciones en el pipeline GitLab, como administrador de tareas Trello y como repositorio de casos de prueba Google Sheets.

Finalmente, para el desarrollo de la práctica se tuvo una primera fase de adaptación al proyecto y al equipo de dos semanas, para luego pasar al ciclo de automatización y mantenimiento descrito anteriormente. Acto seguido se empezaron a automatizar los casos de prueba, teniendo como enfoque completar módulos enteros antes de iniciar otros y así hasta finalmente superar el resultado esperado.

4 Resultados

Tras varios meses de automatización, mantenimiento y trabajo en equipo finalmente se lograron automatizar 7 de 15 módulos en su totalidad, omitiendo por supuesto aquellos casos de prueba que no eran automatizables. Además, se aumentó la cantidad de pruebas automatizadas en 200 aproximadamente, que sumando las que ya existían anteriormente dan un total de 375 pruebas automatizadas. A su vez, esto implicó analizar la viabilidad de automatización de un aproximado de 220 casos de prueba.

En la siguiente tabla (**Tabla 1**) se muestra la cantidad de casos de prueba, los casos de prueba automatizados y los casos de prueba no automatizables de cada uno de los 7 módulos completados.

Tabla 1

Cantidad de casos de prueba de los módulos terminados

Módulo	Casos de prueba	Automatizados	No automatizables
Career Development	17	17	0
Education	14	14	0
Client Portfolio	6	6	0
Finances	59	38	21
Notifications	17	14	3
General	31	31	0
Leadership	6	6	0
Total	150	126	24

Como es visible en la tabla, hubo una cantidad considerable de casos no automatizables dentro del módulo 'Finances', esto fue debido a que estas pruebas estaban relacionadas con la validación de un documento descargable de tipo pdf, archivo que no era posible verificar apropiadamente a través de Playwright, además de otras pruebas asociadas al estilo visual de la página, que tampoco eran plausibles para la automatización.

Además de estos casos de prueba, también hay bastantes ya automatizados repartidos a lo largo de los demás módulos, como podemos ver en la **Tabla 2** donde queda expresada la cantidad de casos de prueba total, casos automatizados, casos pendientes por automatizar y casos no automatizables. Aún queda una gran cantidad de casos de prueba pendientes por automatizar y por validar su viabilidad para ser automatizados.

Tabla 2

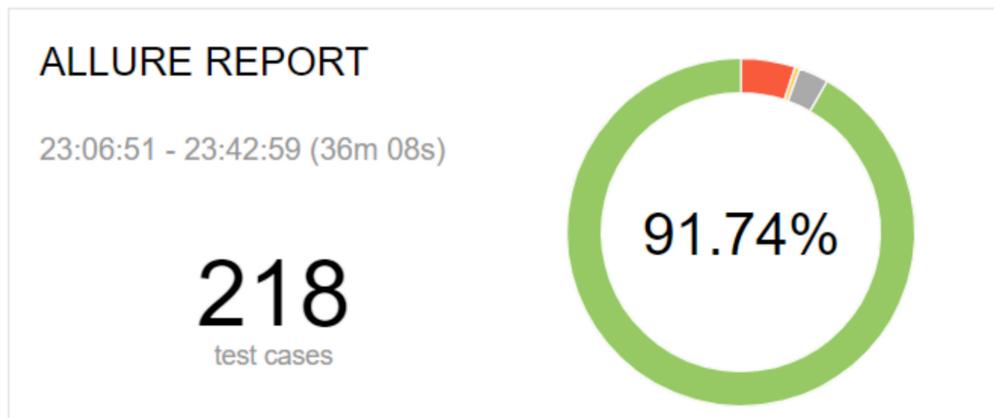
Cantidad de casos de prueba de los módulos no terminados

Módulo	Casos de prueba	Automatizados	Pendientes	No automatizables
Budget Approval	7	5	1	1
Happiness	21	18	2	1
Delivery	157	13	144	0
Resources	82	52	22	8
Admin	154	142	6	6
Evaluation	6	0	6	0
My Dashboard	16	15	1	0
Technology	9	4	5	0
Total	452	249	187	16

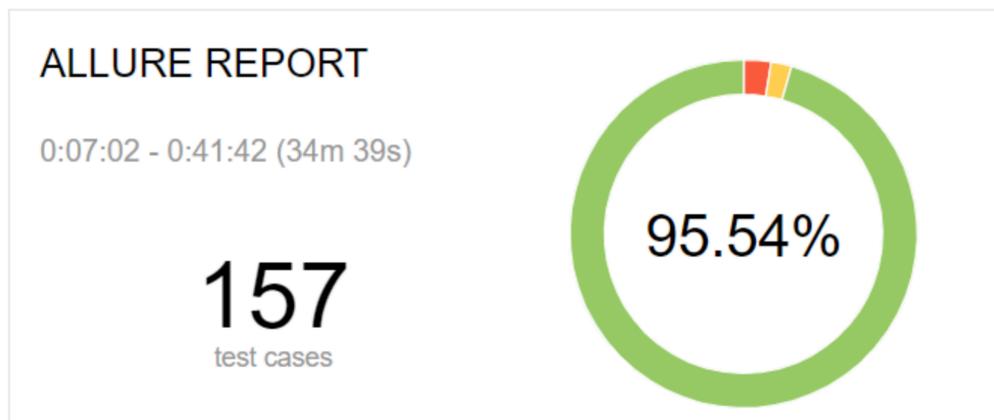
Mientras que, el resultado del mantenimiento ha demostrado ser bastante efectivo manteniendo la meta de un porcentaje menor al 10 % en la tasa de fallos como podemos verificar en las figuras 1 y 2, donde se visualizan los resultados de la ejecución del suite de pruebas que actualmente está dividido en dos partes por motivos relacionados con la duración de la sesión.

Figura 2

Primera parte del resultado de la ejecución del suite de pruebas

**Figura 3**

Segunda parte del resultado de la ejecución del suite de pruebas



En estos resultados podemos visualizar una gráfica circular con el porcentaje de casos de prueba exitosos en el centro, indicándonos que la tasa de fallos fue del 8.26% en 218 casos de prueba y 4.46% en 157 casos de prueba en las figuras 1 y 2 respectivamente.

En cuanto al tiempo usado por el equipo de pruebas manual, que actualmente está compuesto por un único individuo, en lo que corresponde a la ejecución de pruebas de regresión, se estima que el total de 375 pruebas automatizadas ahorró un total de 125 horas de trabajo manual.

Finalmente, cabe destacar que aún hay varios de los módulos cuyos casos de prueba proveen una baja cobertura al mismo, siendo necesaria la intervención del equipo de pruebas manual para finalizar de introducir nuevos casos que la mejoren. También, es necesario señalar que varios de los casos de prueba fallidos actualmente en los resultados obtenidos son debido a bugs por lo que en realidad el porcentaje de falsos negativos es aún menor, demostrando el excelente trabajo que ha realizado el equipo en cuanto al mantenimiento del suite de pruebas automatizadas.

5 Análisis

Los resultados de la automatización de pruebas fueron muy satisfactorios para el proyecto, aunque también destacan la importancia de mantener al personal de pruebas manuales dentro de los equipos de desarrollo o al menos alguien que cumpla sus funciones, pues no se deben olvidar las limitaciones de la automatización de pruebas. Luego de que se analizaran varios de los casos de prueba y se determinara que no eran automatizables, quedó claro que seguramente existen múltiples casos como éste en aplicaciones en desarrollo, donde hay funcionalidades de máxima importancia que no podrán ser automatizadas y tendrá que haber alguien del personal que se encargue de realizar estas pruebas.

Así mismo, la cobertura en las pruebas no siempre es la más indicada, como se pudo evidenciar en algunos de los módulos de este proyecto, por lo que se reitera la importancia de mantener a alguien encargado del trabajo manual de las pruebas, para que éste pueda realizar una exploración más libre dentro de la aplicación, sin limitarse a los casos de prueba escritos, permitiendo encontrar errores que no se hubieran podido descubrir de otra forma.

De igual manera, para que los beneficios de la automatización se vean reflejados de forma más rápida, el trabajo en equipo es esencial en un proyecto tan grande, con tantos módulos y casos de prueba cómo lo es People App, con total seguridad, que los resultados esperados se lograron superar solo gracias a la cantidad de personas que habían dedicadas a la automatización de pruebas, sin considerar las vacaciones, permisos y días feriados que interrumpieron en cierta medida el flujo de trabajo.

Finalmente, la estabilidad de la aplicación juega un papel muy importante para iniciar con cualquier proceso de automatización, pues ésta tiene una implicación directa con el esfuerzo que se debe invertir en mantenimiento. Una aplicación en constante cambio o poco refinada requerirá de una corrección de pruebas constante, restando tiempo y esfuerzo al aumentar la cobertura de la aplicación con más pruebas automatizadas para que así pueda beneficiarse significativamente de las ventajas que ofrece esta grandiosa práctica.

6 Conclusiones

Considerando el objetivo general del proyecto y sus resultados, se puede concluir que éste se cumplió exitosamente, pues la implementación de pruebas automatizadas dentro del proyecto logró disminuir significativamente el tiempo invertido en la ejecución de pruebas manuales, dejando claro que el objetivo principal del proyecto se cumplió exitosamente a pesar del largo tramo que aún queda por recorrer dentro del proyecto.

Dentro de los logros más destacados de la implementación de pruebas automatizadas en el proyecto People App están:

- Analizar la viabilidad de automatización de 220 casos de prueba.
- Aumentar los casos de prueba automatizados en un aproximado de 200, dejando un total de 375 pruebas automatizadas.
- Mantener una tasa de fallos menor al 10 % en la ejecución del suite de pruebas automatizadas, aún teniendo bugs presentes en la aplicación.
- Reducir el tiempo de ejecución de pruebas manuales en 125 horas aproximadamente, que finalmente se tradujo en la realización de entregas más frecuentes.

Esto representa un rotundo éxito respecto a los objetivos planteados para la realización de este proyecto de automatización dentro de People App, que no hubiese sido posible sin el trabajo en equipo realizado a través de su desarrollo.

7 Recomendaciones

Las próximas acciones que recomendaría realizar en el proyecto People App, son inicialmente aumentar el personal de pruebas manuales para que pueda actualizar el repositorio de casos de prueba, aumentando la cobertura de algunos de sus módulos y corrigiendo varios casos de prueba que podrían estar deprecados actualmente. En caso de que no se pueda aumentar el personal, asignar a alguno de los integrantes del equipo de automatización al equipo de pruebas manuales para que ayude con esta importante tarea.

Segundo y último, seguir con el buen trabajo que se ha estado realizando, aumentando la cantidad de casos de prueba automatizados y manteniendo una tasa baja de fallos en las ejecuciones diarias, y tal vez, cuando tanto la aplicación como el suite de pruebas automatizadas sean mucho más estables, considerar la unificación del pipeline de pruebas automatizadas al pipeline de despliegue de la aplicación si podría llegar a traer algún beneficio en el caso específico de People App.

Referencias

- ISTQB. (2023, May 10). *ISTQB Glossary*. Syllabus - Version 3. <https://glossary.istqb.org>
- Jennifer Castro. (2021, September 7). *What is QA Automation and Is it Right for Your Team?* Growth Acceleration Partners. <https://bit.ly/47jCq7a>
- Playwright. (2023). *Playwright enables reliable end-to-end testing for modern web apps*. <https://playwright.dev>
- Sarah Laoyan. (2022, August 17). *Agile Manifesto: la guía para entender la metodología Agile*. Asana. <https://bit.ly/48Ibglh>
- Sarah Laoyan. (2022, October 31). *Scrumban: lo mejor de dos metodologías ágiles*. Asana. <https://bit.ly/48X4fU7>

Anexos

Anexo 1. Propuesta del proyecto de práctica

En este documento se pueden encontrar los resultados esperados a los que se suele hacer referencia en algunas partes del presente texto.

- Propuesta del proyecto de práctica: <https://bit.ly/3vnzWHs>