



**Informe de Prácticas Profesionales como
Opción de Grado**



**IMPLEMENTACIÓN DE UN PROTOCOLO PARA LA AUTOMATIZACIÓN DE
PRUEBAS POR MEDIO DEL FRAMEWORK CYPRESS PARA DESARROLLOS
DE SOFTWARE EN AGILE DEVOPS.**

PRESENTADO POR:

CRISTIAN SÁNCHEZ ARIÑA

**Código:
2016114123**

PRESENTADO A:

**Ing. LUIS ALBERTO PAVA CARMONA
Tutor de prácticas profesionales**

**OSCAR CIFUENTES OSPINA
Tutor Empresarial**

**UNIVERSIDAD DEL MAGDALENA
FACULTAD DE INGENIERÍA
INGENIERIA DE SISTEMAS
SANTA MARTA D.T.C.H
24/12/2021**



Informe de Prácticas Profesionales como Opción de Grado



Tabla de Contenido

1. PRESENTACIÓN.....	4
2. OBJETIVOS Y/O FUNCIONES.....	5
2.1. Objetivo General:.....	5
2.2. Objetivos Específicos:.....	5
2.3. Funciones del practicante en la organización:.....	5
3. JUSTIFICACIÓN:.....	9
4. GENERALIDADES DE LA EMPRESA:.....	10
5. SITUACIÓN ACTUAL.....	15
6. BASES TEÓRICAS RELACIONADAS.....	17
7. DESARROLLO DE ACTIVIDADES:.....	21
8. CRONOGRAMA:.....	33
9. CONCLUSIONES Y LÍNEAS FUTURAS.....	34
10. BIBLIOGRAFÍA.....	35



Informe de Prácticas Profesionales como Opción de Grado



Tabla de Ilustraciones.

Ilustración 1 Codificación de pruebas funcionales.....	6
Ilustración 2 Monitoreo de pruebas funcionales en Azure Devops.....	7
Ilustración 3 Reporte de bugs en Azure Devops	8
Ilustración 4 Sede Principal de KPMG Colombia	11
Ilustración 5 Organigrama Agile Devops	13
Ilustración 6 Metodología de trabajo en Agile Devops	14
Ilustración 7 Descripción detallada del proceso y los subprocesos seleccionados para aplicar el trabajo en KPMG	14
Ilustración 8 Instalación de cypress mediante la consola del Visual Studio Code.	24
Ilustración 9 Carpetas del framework creadas al instalarse	24
Ilustración 10 Ventana de Cypress.....	26
Ilustración 11 Simulación de ejecución de una prueba realizada exitosamente con el framework Cypress.....	27
Ilustración 12 Codificación de los scripts en Visual Studio Code.	28
Ilustración 13 Historial de pruebas realizadas vista en el entorno del framework Cypress	29
Ilustración 14 Pantalla de visualización de resultado de las pruebas en cypress .	30
Ilustración 15 Descripción general de las pruebas realizadas.....	30
Ilustración 16 Visualización de resultados de las pruebas en la terminal de Visual Studio Code.....	31
Ilustración 17 Cronograma general de actividades	33



Informe de Prácticas Profesionales como Opción de Grado



1. PRESENTACIÓN

El presente informe se centra en el desarrollo del proyecto realizado durante las prácticas profesionales en la empresa KPMG en el área de Agile DevOps, el cual consiste en el desarrollo de un sistema de pruebas automatizadas, es una innovación total dentro del área de trabajo y que tendrá un impacto positivo debido a que ayuda a optimizar los procesos de pruebas y los proyectos que se están llevando a cabo en el área de Agile DevOps presenten menos retrasos en la entrega a los clientes, optimice el desarrollo de actividades y se pueda compartir los resultados de dichas pruebas de forma ágil de tal manera que todos los componentes del equipo de desarrollo los puedan conocer.

Además, se destacan mis funciones en la empresa como practicante, las principales generalidades de esta, y en cuanto a la realización del proyecto más allá de lo dicho en el presente párrafo explicare las metodologías que adopte para realizar este, los objetivos de este, las razones por las cuales decidí desarrollarlo, el alcance que tendrá y unas conclusiones generales sobre cómo podría mejorar o actualizar el proyecto a futuro y en general de las experiencias adquiridas en este periodo.



2. OBJETIVOS Y/O FUNCIONES

2.1. Objetivo General:

Implementar un protocolo de pruebas automatizadas de software a través del framework Cypress, que permita que los aplicativos desarrollados sean óptimos, se reduzcan errores y se controle mejor el desarrollo de pruebas QA en el área Agile DevOps.

Defina el objetivo general del proyecto

2.2. Objetivos Específicos:

1. Analizar y seleccionar los casos de prueba a automatizar para las aplicativos de software de Agile DevOps
2. Diseñar un protocolo de pruebas relacionado con las historias de usuario de los distintos proyectos donde se vaya a aplicar la respectiva automatización en Agile DevOps
3. Desarrollar un plan de pruebas automatizadas para los aplicativos de software de Agile DevOps
4. Configurar y ejecutar los casos de pruebas planteados para cada aplicativo desarrollado en Agile DevOp

2.3. Funciones del practicante en la organización:

1. Desarrollo de pruebas funcionales y no funcionales (Testing) tanto para proyectos informáticos internos como proyectos y aplicaciones web que requieran los clientes de la empresa mediante la automatización de estas.



Informe de Prácticas Profesionales como Opción de Grado



```
describe('cuentas',function () {
  1
  2
  3
  4
  5
  6
  7
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  it('login y primeras preguntas', function () {
    cy.visit('http://cobogweb14qa/login')
    cy.get('[placeholder="Correo Electrónico"]').type('semunoz@kpmg.com')
    cy.get('[placeholder="Contraseña"]').type('123')
    cy.contains('Ingresar').click()
    cy.get('a > .pl1-file-text-image').click()
    cy.contains(' Completa las preguntas de este formulario')
    cy.contains('¿En los últimos 15 días te han realizado la prueba para determinar si tienes COVID-19?')
    cy.get('#tab-1 > :nth-child(1) > .col-sm-12 > .form-control').select('No')
    cy.get('#tab-1 > :nth-child(4) > .col-sm-12 > .form-control').select('No')
    cy.get('#tab-1 > :nth-child(5) > .col-sm-12 > .form-control').select('Si')
    cy.get('#tab-1 > :nth-child(6) > .col-sm-12 > .form-control').select('primera dosis')
    // cy.get('#file').attachFile('images/archivo.png');
    cy.get('.next')
  })

  it('adjuntar archivo', function () {
    //cy.get('#file_question').attachFile('images/archivo.png');
    cy.get(':nth-child(7) > .col-sm-12 > .input-group > .form-control').attachFile('images/archivo.png');
    cy.get('#tab-1 > :nth-child(8) > .col-sm-12 > .form-control').select('No')
  })
})
```

Ilustración 1 Codificación de pruebas funcionales

El proceso de automatización de las pruebas lo ejecuto programando scripts en el lenguaje JavaScript usando el framework de pruebas automatizadas Cypress como se aprecian en la anterior imagen

2. Acompañamiento y Participación en metodologías agiles para el aseguramiento de la calidad (SCRUM). (Dailys, Reviews, Release, Reunión de pruebas funcionales con los usuarios finales)
3. Monitoreo y desarrollo de pruebas a través de Azure Devops (repositorio) de las aplicaciones y proyectos que se estén llevando a cabo.

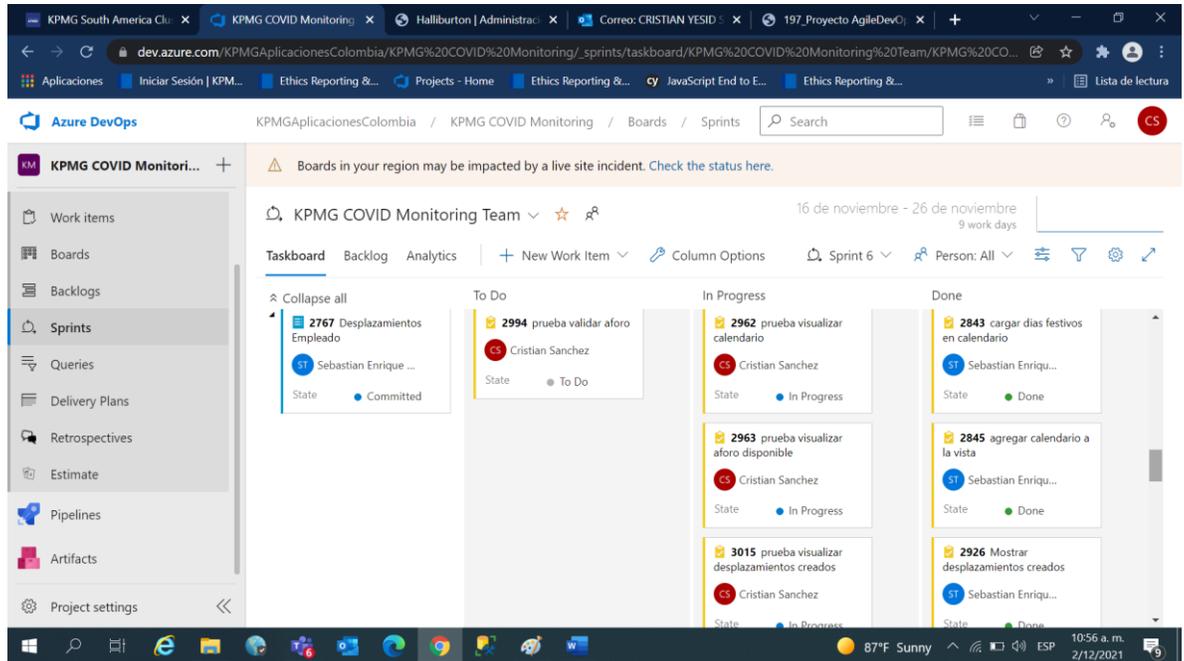


Ilustración 2 Monitoreo de pruebas funcionales en Azure DevOps

En la imagen se aprecia el monitoreo que realizo el cual consiste en la creación de tareas en cada historia de usuario que contienen los sprints asociados a cada proyecto en el que me encuentre trabajando, dichas tareas son las pruebas que ejecutare en cada funcionalidad existente en el aplicativo.

4. Reporte de Bugs u otros problemas que se presenten en el desarrollo de las pruebas de los softwares.



Informe de Prácticas Profesionales como Opción de Grado

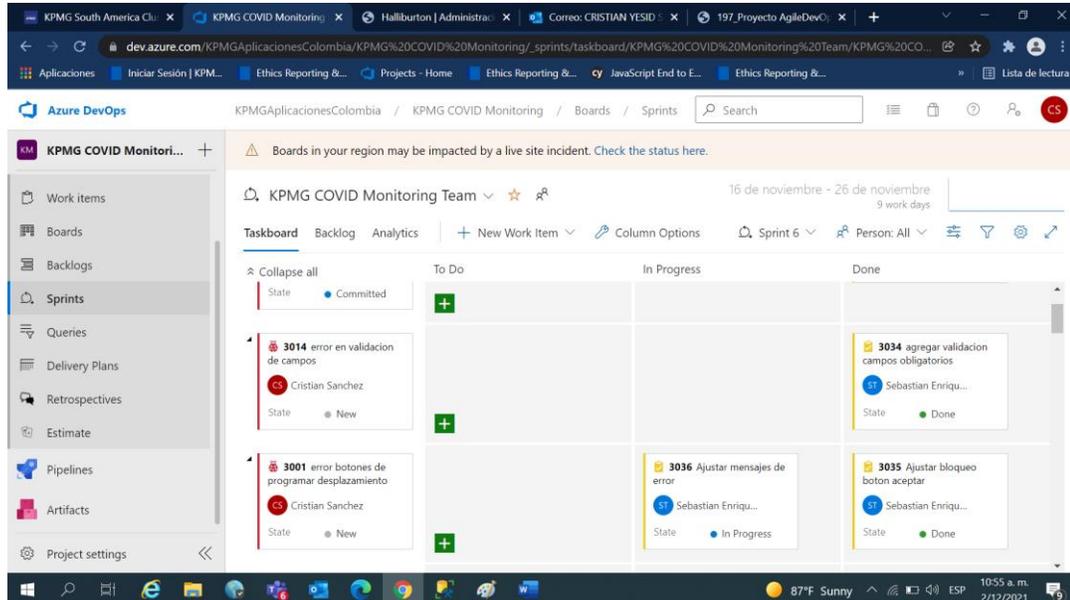


Ilustración 3 Reporte de bugs en Azure Devops

En la anterior imagen se aprecia que por cada bug que se encuentre al realizar las pruebas en alguna funcionalidad del aplicativo se deberá reportar en el Azure Devops con una descripción de este para que el equipo de trabajo pueda reconocerlo y trabajar para solucionarlo.



3. JUSTIFICACIÓN:

Debido a la alta importancia y nivel de complejidad de los proyectos que realiza Agile DevOps implica que se deban realizar una gran cantidad de pruebas para poder abarcar y probar todos los casos de uso de los aplicativos y que estos salgan a producción sin fallos o bugs es crucial empezar a optimizar los procesos de testing en el área.

Con la solución propuesta aquí, el desarrollo y en especial la solución de bugs que se encuentren en los proyectos será mucho más óptima, esto hará que se reduzca gran medida el trabajo y que además se hagan pruebas más recurrentes en los aplicativos lo que aumenta la detención de errores en los aplicativos y mejorara la usabilidad de este.

El presente proyecto, por una parte, intentara optimizar en gran medida los procesos de QA para las soluciones de software Agile DevOps, reduciendo el tiempo de ejecución para pruebas funcionales repetitivas, permitiendo que las pruebas manuales solo se ejecuten en escenarios especiales, detectar bugs de manera más rápida y detectar bugs en etapas tempranas del desarrollo para que el impacto de estos no frene las actividades o que puedan llegar a un ambiente productivo frenando operaciones de los usuarios. Por otra parte, se espera ejecutar una mayor cantidad de pruebas, con mayor precisión, sin necesidad de incrementar el número de personas dedicadas al momento de validar escenarios, generando así que el equipo de trabajo sea mucho más eficiente y productivo, aumentando más la rentabilidad de la ejecución de las pruebas de software y de los desarrollos.

Estas pruebas automatizadas se sugieren realizarse con cada cambio de versión de los aplicativos.



Informe de Prácticas Profesionales como Opción de Grado



4. GENERALIDADES DE LA EMPRESA:

KPMG es una red global de firmas de servicios profesionales fundada en 1911, que ofrece servicios de auditoría, de asesoramiento legal y fiscal, y de asesoramiento financiero y de negocio en 156 países. Es una de las cuatro firmas más importantes del mundo de servicios profesionales y cuenta en la actualidad con 197.263 empleados trabajando en firmas miembro a nivel mundial.

Dentro de KPMG Colombia existe Agile Devops, una célula que se encuentra en el área de consultoría y que se dedica al desarrollo de proyectos de software que consisten en la implementación de aplicativos webs para procesos internos o de clientes de la firma y también a la mejora de procesos a través de la analítica de datos.

Nuestro objetivo central es el de desarrollar soluciones tecnológicas que apalanquen la estrategia comercial de las diferentes líneas de negocio de la Firma, con el fin de lograr un mayor reconocimiento y posicionamiento en el mercado.

Breve historia

En 1911 William Barclay Peat & Co. Y Marwick Mitchell & Co. se unieron para formar lo que más tarde se conoció como Peat Marwick International (PMI). Klynveld Kraayenhof & Co. Se fusionó en 1979 con la Deutsche Treuhand Gesellschaft y la firma internacional McLintock Main Lafrentz & Co. para formar KMG. Finalmente, en 1987 KPMG fue constituida por la fusión de KMG y PMI. KPMG, comenzó operaciones en Colombia hace más de 60 años, prestando servicios de Auditoría, Impuestos, Consultoría y Legal.

Hoy en día atendemos cerca de 1.300 clientes, nacionales y extranjeros, pertenecientes a los sectores productivos: banca, seguros, manufactura y comercio, sector privado y del Gobierno. Nos acompañan más de 1.600 profesionales: Contadores, Administradores de Empresas, Economistas, Abogados, Ingenieros Industriales, Ingenieros de Sistemas, seleccionados cuidadosamente. Contar con



Informe de Prácticas Profesionales como Opción de Grado



una sola entidad de servicios complementarios puede brindarle una mejor relación beneficio costo.

Ubicación

KPMG tiene su sede central en la ciudad de Amstelveen en Holanda mientras que KPMG Colombia tiene su sede principal en la ciudad de Bogotá y cuenta con otras sedes en las ciudades de Cali, Medellín, Bucaramanga Y Barranquilla.

La sede central en Colombia está ubicada en la Calle 90, #19c74 barrio chico en la ciudad de Bogotá.

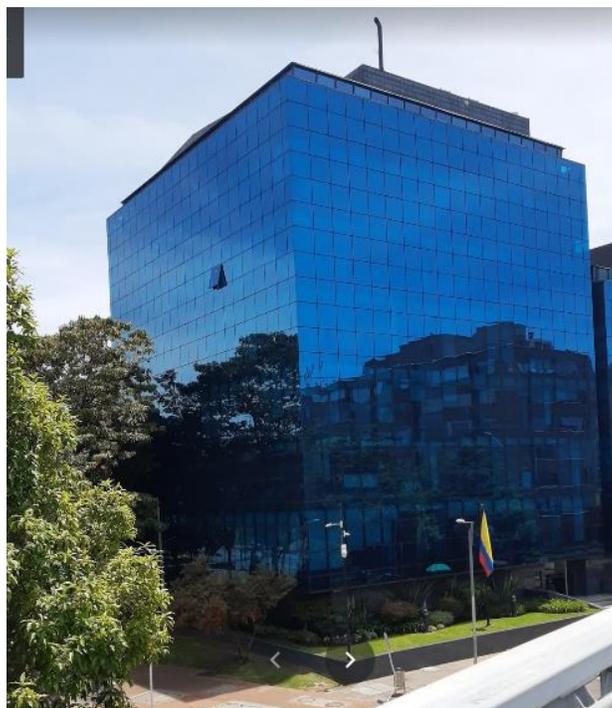


Ilustración 4 Sede Principal de KPMG Colombia

Edificio de KPMG en la ciudad de Bogotá, Colombia

Misión

Dar respuesta a los principales retos de la sociedad en Colombia a través de nuestra capacidad de generar impacto y transformación social.

	Informe de Prácticas Profesionales como Opción de Grado	
---	--	---

Visión

Aportar confianza e impulsar la transformación de la sociedad a través de la actuación de nuestros profesionales, con el fin de apoyar la consecución de los Objetivos de Desarrollo Sostenible y promover un futuro mejor para todos.

Valores de la firma:

Nuestros Valores crean un sentido de identidad compartida. Son 5 palabras simplificadas que expresan de una manera más poderosa y clara quienes somos, en lo que creemos y lo que defendemos, a través de las elecciones individuales que hacemos todos los días. También nos acercan como organización global. Nuestros valores son las raíces de nuestro actuar y nos sirven como guía en nuestro viaje cultural.

- **INTEGRITY** - Integridad: Hacemos lo correcto
- **EXCELLENCE** - Excelencia: Nunca dejamos de aprender y mejorar.
- **COURAGE** - Valentía: Pensamos y actuamos con valor
- **TOGETHER** - Juntos: Nos respetamos y fortalecemos a partir de nuestras diferencias.
- **FOR BETTER** - Ser Mejores: Hacemos lo que es importante.



Informe de Prácticas Profesionales como Opción de Grado



Organigrama Agile DevOps KPMG Colombia

[Organigrama](#) Directorio

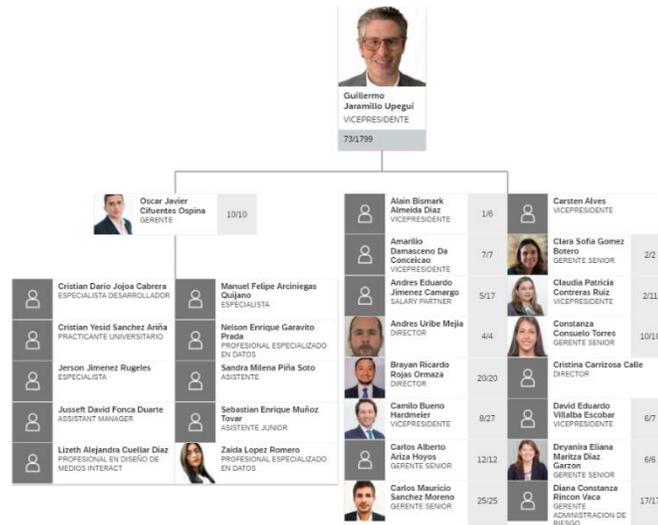


Ilustración 5 Organigrama Agile DevOps

La célula Agile DevOps pertenece a el área de Advisory y consultoría de KPMG Colombia, está compuesta por 10 empleados que se especializan en desarrollo y analítica los cuales trabajamos en el desarrollo de proyectos internos y para clientes de la firma, la cual está a cargo de un gerente general, y el área a su vez esta supervisada por la vicepresidencia de la empresa.

En Agile DevOps manejamos un enfoque caracterizado por el trabajo colaborativo, continuas sincronizaciones y entregas de producto con valor temprano que permiten adaptarnos a los cambios que se presenten en el entorno.

En el área de desarrollo de Agile DevOps en KPMG, la ejecución de pruebas funcionales se realiza de forma manual, este proceso se realiza reportando los bugs y demás fallos encontrados en los aplicativos mediante comunicación directa con el desarrollador y a través de la herramienta Azure Devops en la que se reporta el bug o el fallo mediante la adición de una tarea nueva. Dicho proceso en ocasiones termina siendo no tan eficiente pues se pueden se omitan detalles específicos del

fallo o bug encontrado. Debido a la alta importancia y nivel de complejidad de los proyectos que realiza Agile DevOps implica que se deban realizar una gran cantidad de pruebas para poder abarcar y probar todos los casos de uso de los aplicativos, esto hace que en muchas ocasiones solo se puedan realizar pruebas a las funcionalidades principales y prioritarias para los clientes, dejando de lado algunos bugs que el usuario final pueda llegar a encontrar.

Línea de tiempo, donde se define salidas a producción y entrega de producto con valor.

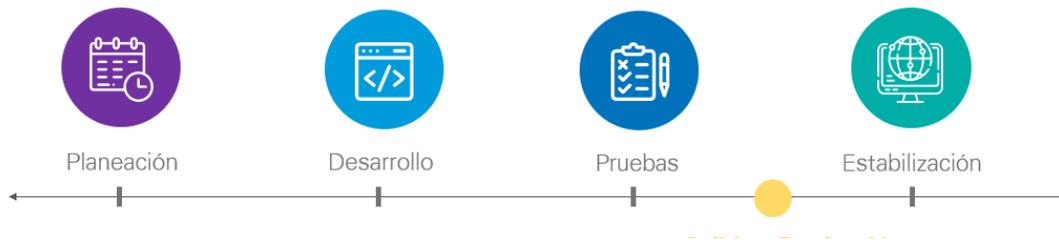


Ilustración 6 Metodología de trabajo en Agile Devops



Ilustración 7 Descripción detallada del proceso y los subprocesos seleccionados para aplicar el trabajo en KPMG



5. SITUACIÓN ACTUAL

Actualmente en los desarrollos de software del área de Agile DevOps se ejecutan pruebas funcionales y no funcionales en los aplicativos que se están realizando. Dichas pruebas son muy importantes y se realizan de forma manual, lo cual hace que el encargado de QA deba invertir más tiempo e inclusive en muchos casos cuando el fallo es grande, involucrar muchas personas del equipo de trabajo, Además se debe tener en cuenta que al hacer pruebas de esta forma hay algunos bugs o fallos que llegan a pasar por alto del QA debido a las pruebas repetitivas y la magnitud de funcionalidades de los proyectos, generando que en ocasiones los usuarios clientes que realizan las pruebas antes de salir a producción reporten fallos y bugs que pueden llegar a tener un impacto en el desarrollo de algunas de las funcionalidades que tienen los distintos aplicativos.

Ejecución de Pruebas AgileDevops

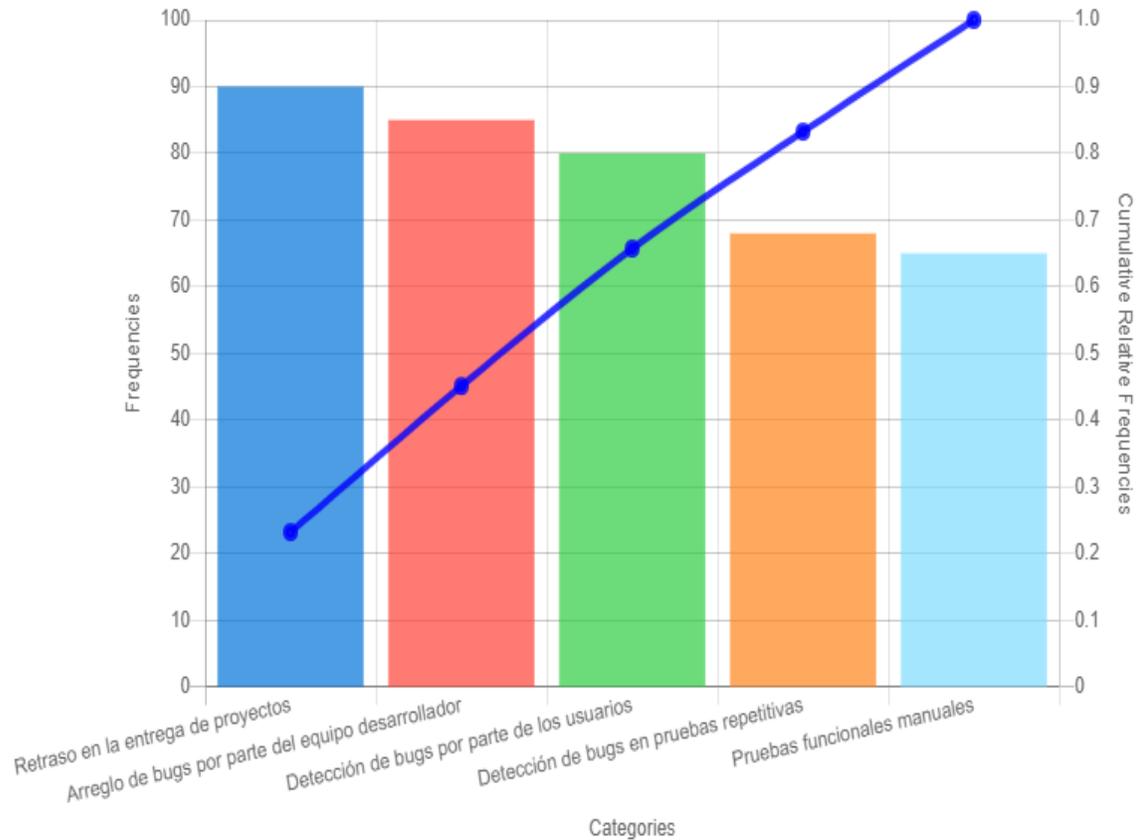


Tabla 1 Representación del proceso de pruebas en Agile Devops

En el anterior diagrama podemos apreciar que el retraso en la entrega de proyectos a los clientes es demasiado alta principalmente porque en la fase anterior al paso a producción la cual es la de pruebas funcionales realizadas por los mismos clientes finales se observa que estos detectan algunos bugs los cuales la mayoría son pasados por alto en las pruebas manuales realizadas por el QA pues están no abarcan todos los aspectos del aplicativos porque al ser repetitivas se prueban contadas funcionalidades en lugar de realizar una prueba general de todo el aplicativo.



6. BASES TEÓRICAS RELACIONADAS

Automatización de pruebas: En las pruebas de software, la automatización de pruebas consiste en el uso de software especial para controlar la ejecución de pruebas y la comparación entre los resultados obtenidos y los resultados esperados.

Cypress: Cypress es un framework de pruebas automatizadas “todo en uno” que incluye librerías de aserciones, de mocks y pruebas e2e automáticas sin utilizar Selenium. Cypress no utiliza Selenium porque consta de una nueva arquitectura, construida desde cero, que ejecuta los comandos en el mismo ciclo de ejecución que la aplicación. Detrás de Cypress se ejecuta un proceso Node que constantemente se comunica, sincroniza y ejecuta tareas, teniendo acceso tanto a la parte front como a la parte back de la aplicación y respondiendo a los eventos en tiempo real. Esta herramienta está diseñada especialmente para manejar frameworks de JavaScript modernos, React, Angular, Vue, Elm, etc. Pero, también funciona igual de bien en páginas o aplicaciones renderizadas en servidor.

Framework: Es un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Ingeniera de software: La ingeniería del software es una disciplina que implica el uso de estructuras, herramientas y técnicas para construir programas



Informe de Prácticas Profesionales como Opción de Grado



informáticos. Asimismo, incluye el análisis previo de la situación, la redacción del proyecto, la creación del software y las pruebas necesarias para garantizar su correcto funcionamiento antes de que el sistema esté operativo. Esta ingeniería aborda todas las fases del ciclo de vida de desarrollo de cualquier tipo de sistema de información y es aplicable a una amplia gama de ámbitos de la informática y la ciencia de los ordenadores, como el diseño de compiladores, sistemas operativos y tecnologías de Intranet/Internet: la empresa, la investigación científica, la medicina, la fabricación, la logística, la banca, el control del tráfico y la meteorología son solo algunos de los campos en los que puede trabajar.

JavaScript: JavaScript es un lenguaje de programación poderoso, capaz de aportar soluciones eficaces en la mayoría de los ámbitos de la tecnología.

Es especialmente importante porque es el único lenguaje de programación que entienden los navegadores, con el que se desarrolla la parte de la funcionalidad front-end en sitios web y aplicaciones web modernas. Pero también es fundamental en muchos otros tipos de desarrollos. Sus usos más importantes son los siguientes:

Desarrollo de sitios web del lado del cliente (frontend, en el navegador)

Desarrollo de todo tipo de aplicaciones gracias a la plataforma NodeJS

Desarrollo de aplicaciones para dispositivos móviles, híbridas o que compilan a nativo

Desarrollo de aplicaciones de escritorio para sistemas Windows, Linux y Mac, pudiendo escribir un código compatible con todas las plataformas.

Por tanto, podemos considerar a Javascript el lenguaje universal, pues es el que más tipos de aplicaciones y usos que puede abarcar en la actualidad.

Node.js: es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el



Informe de Prácticas Profesionales como Opción de Grado



lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web.

QA: El aseguramiento de la calidad (se usa con frecuencia el anglicismo quality assurance, QA) es el conjunto de actividades planificadas y sistemáticas aplicadas en un sistema de gestión de la calidad para que los requisitos de calidad de un producto o servicio sean satisfechos. Entre estas actividades se encuentran la medición sistemática, la comparación con estándares, el seguimiento de los procesos, todas actividades asociadas con bucles de realimentación de información. Estas actividades contribuyen a la prevención de errores, lo cual se puede contrastar con el control de calidad, que se centra en las salidas del proceso. Ambos conceptos suelen utilizarse de manera conjunta

Programación web: Permite la creación de sitios dinámicos en Internet. Esto se consigue generando los contenidos del sitio a través de una base de datos mediante lenguajes de programación Web. Dominando la programación Web podremos crear sitios dinámicos como periódicos digitales o tiendas virtuales, etc.

Scrum: La metodología Scrum es un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos. Es decir, se trata de una metodología de trabajo ágil que tiene como finalidad la entrega de valor en períodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación. Esto permite al cliente, junto con su equipo comercial, insertar el producto en el mercado pronto, rápido y empezar a obtener ventas.



Informe de Prácticas Profesionales como Opción de Grado



Sistema Gestor de Bases de datos: Un sistema gestor de base de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos. Los usuarios pueden acceder a la información usando herramientas específicas de consulta y de generación de informes, o bien mediante aplicaciones al efecto. Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

Testing: El Testing de Software es toda una disciplina en la ingeniería de software permite tener procesos, métodos de trabajo y herramientas para identificar defectos en el software alcanzando un proceso de estabilidad del mismo. El Testing no es una actividad que se piensa al final del desarrollo del software, va paralelo a este. Permite que lo que se está construyendo, se realice de manera correcta de acuerdo a lo que necesita un usuario final. De ahí radica su importancia, pues es una forma de prevenir o inclusive de corregir posibles desviaciones del software antes de que sea operable. Se tenía la equivocada idea que el testing se realizaba al final, cuando ya el software estaba codificado y justo antes de entregarlo a la operación, pero actualmente el testing de software debe ir desde el inicio del proceso.

Visual Studio Code: Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, MacOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.



7. DESARROLLO DE ACTIVIDADES:

PROTOCOLO DE PRUEBAS AUTOMATIZADAS

A continuación, se presenta la estrategia para la realización de pruebas automatizadas, se describe el alcance de las pruebas, el ambiente de pruebas, los recursos necesarios, las herramientas a utilizar, los riesgos, planes de contingencia, inicialmente enumeraremos las contemplaciones que tuvimos en la siguiente tabla:

Contemplaciones que se tuvieron en cuenta en el proceso
1. Alcance de las pruebas automatizadas
2. Riesgos
3. Ambiente de pruebas
4. Herramientas necesarias
5. Criterios de entrada y salida
6. Reporte de pruebas

Tabla 2 Factores que se tuvieron en cuenta para realizar el proyecto

1. ALCANCE DE PRUEBAS LAS AUTOMATIZADAS

El protocolo de pruebas tiene como objeto general establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación de cada una de las pruebas automatizadas planeadas para la solución seleccionada.

2. RIESGOS

En la ejecución de las pruebas automatizadas, los riesgos que afectaran pueden ser reiterativos:

- Falta de recursos para la ejecución de las pruebas.
- Tiempo reducido asignado a la fase de pruebas.
- Cambios frecuentes en la definición de las funcionalidades.
- Falta de coordinación entre los equipos de desarrollo y testing.
- Funcionalidades no terminadas en el tiempo estimado.



Informe de Prácticas Profesionales como Opción de Grado



- Solicitud de cambios en aquellas funcionalidades que ya tienen casos de pruebas automatizados.

3. AMBIENTE DE PRUEBAS

Ambiente donde se ejecutarán cada una de las pruebas:

- Navegadores Chrome, Mozilla Firefox, Microsoft Edge
- Sistemas Operativos Windows, Windows server
- Ambiente de QA, Sobre los servidores de QA proporcionados

4. HERRAMIENTAS NECESARIAS

Herramientas necesarias para la creación y ejecución de cada una de las pruebas:

- Cypress: Framework de automatización seleccionado
- Node.js: Creación de la estructura del proyecto, uso e importación de librerías
- Visual Studio Code IDE: seleccionado para la programación de las pruebas

5. CRITERIOS DE ENTRADA Y SALIDA

5.1 Criterios de entrada

- Las funcionalidades deben estar en el ambiente de QA con una primera autorización del desarrollador para ser probadas.
- Servidor y ambiente de QA disponibles.
- Se hará una primera revisión de la prueba a automatizar de manera manual.
- El framework de automatización debe estar instalado y listo para la ejecución de pruebas.
- En caso de encontrarse algún bug de manera manual este ya debe estar solucionado y cerrado.



5.2 Criterios de salida

- Ejecución de todas las pruebas automatizadas.
- Cobertura de las historias y requerimientos funcionales necesarios bajo las pruebas.
- Ningún bug de gran medida encontrado.

6. REPORTE DE PRUEBAS

El reporte de pruebas se obtendrá automáticamente mediante el framework Cypress. Este reporte informara el resultado de cada una de las pruebas, casos fallidos y exitosos, el tiempo de ejecución tomado por cada prueba y en su totalidad. Incluirá el paso a paso de cada una de las pruebas, mediante un video que realizará el mismo framework, además con la herramienta debug con que cuenta el framework se podrá ver el resultado de una manera óptima.

DESARROLLO DE LA AUTOMATIZACION DE PRUEBAS

Para el desarrollo de las pruebas automatizadas se tuvieron en cuenta los siguientes pasos enumerados cronológicamente en la siguiente tabla:

Pasos para llevar a cabo el desarrollo de la automatización de pruebas
1. Instalación y preparación
2. Estructuración de las pruebas
3. Configuración del entorno de pruebas
4. Ejecución del plan de pruebas
5. Resultado y reporte de las pruebas

Tabla 3 Paso a Paso del proyecto

1. INSTALACIÓN Y PREPARACIÓN

Para la construcción del proyecto se empieza con la estructuración usando Node.js que nos permitirá descargar un proyecto Cypress desde cero en el directorio deseado, por buenas prácticas se recomienda que el archivo

package.json ya este creado para verificar que el proyecto se descargue en la dirección correcta.

Una vez terminada la descarga del proyecto, tendrá en la dirección asignada todas las carpetas, con paquetes y librerías del nuevo proyecto Cypress.

```

PS C:\Users\Cristian\Desktop\cypress\test> npm install cypress --save-dev
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
> cypress@9.1.1 postinstall C:\Users\Cristian\Desktop\cypress\test\node_modules\cypress
> node index.js --exec install

Cypress 9.1.1 is installed in C:\Users\Cristian\AppData\Local\Cypress\Cache\9.1.1

npm WARN test@1.0.0 No repository field.

+ cypress@9.1.1
updated 1 package and audited 164 packages in 26.839s
found 0 vulnerabilities
  
```

Ilustración 8 Instalación de cypress mediante la consola del Visual Studio Code

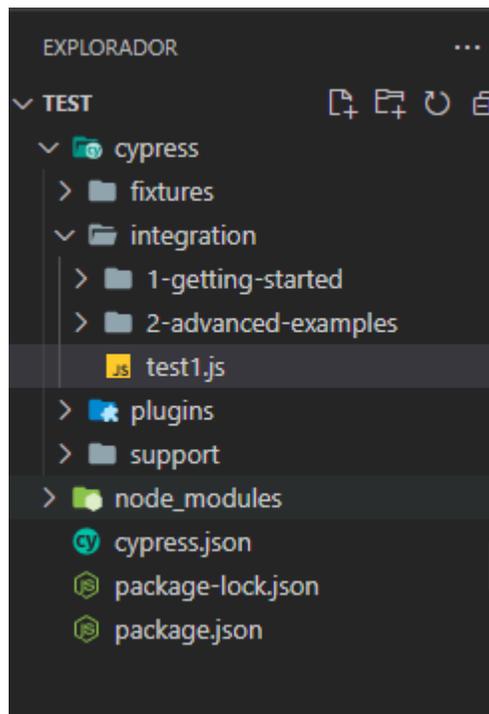


Ilustración 9 Carpetas del framework creadas al instalarse

Con el proyecto correctamente descargado e instalado, podemos iniciar con el desarrollo de nuestros primeros scripts, para esto crearemos un archivo con la extensión “. spec.js” dentro de la carpeta “integration”. Cypress tomara los archivos dentro de esta carpeta como las pruebas que se desean ejecutar.

	Informe de Prácticas Profesionales como Opción de Grado	
---	--	---

Para este proyecto se creó un total de 3 archivos con extensión “.js” con pruebas, por cada una de las funcionalidades que se determinó automatizar durante la creación del plan de pruebas.

2. ESTRUCTURACION DE LAS PRUEBAS

Con Cypress podremos imitar cada una de las interacciones que realizara el usuario con la aplicación de manera sencilla a través de comandos que obtienen elementos DOM (Document Object Model en español Modelo de Objetos del Documento) de nuestra página web o realizan acciones sobre navegador web que se esté utilizando sin la necesidad de algún controlador o programa externo que intervenga entre el código de la prueba y el navegador.

Los comandos más utilizados son:

- **Cy.get:** Obtiene uno o más elementos DOM por un selector o alias.
- **Cy.contains:** Obtiene el elemento DOM que contenga un texto seleccionado, los elementos DOM que se deseen obtener pueden tener más del texto deseado y aun así coincidir.
- **Cy.visit:** visita una URL
- **Cy.should:** Crea afirmaciones, que se reintentan automáticamente hasta que encuentra una respuesta o se agota el tiempo de espera. Comando importante ya que permite hacer las afirmaciones sobre los elementos DOM obtenidos mediante los demás comandos.
- **Cy.invoke:** Permite invocar una función previamente establecida o una función con la que cuente el elemento DOM obtenido.
- **Cy.request:** Realiza una petición HTTP.

Con estos métodos ya podemos empezar a escribir nuestras primeras pruebas.

3. CONFIGURACIÓN DEL ENTORNO DE LAS PRUEBAS

Una vez realizado todo el código de cada una de las pruebas se ejecutará cada una de ellas mediante la aplicación para esto es necesario abrir Cypress mediante la consola con el comando “npx cypress open” esto iniciara nuestro proyecto y podremos ver la pantalla de inicio con nuestras pruebas y unas pruebas de ejemplo que ya el mismo Cypress descargo previamente.



Informe de Prácticas Profesionales como Opción de Grado

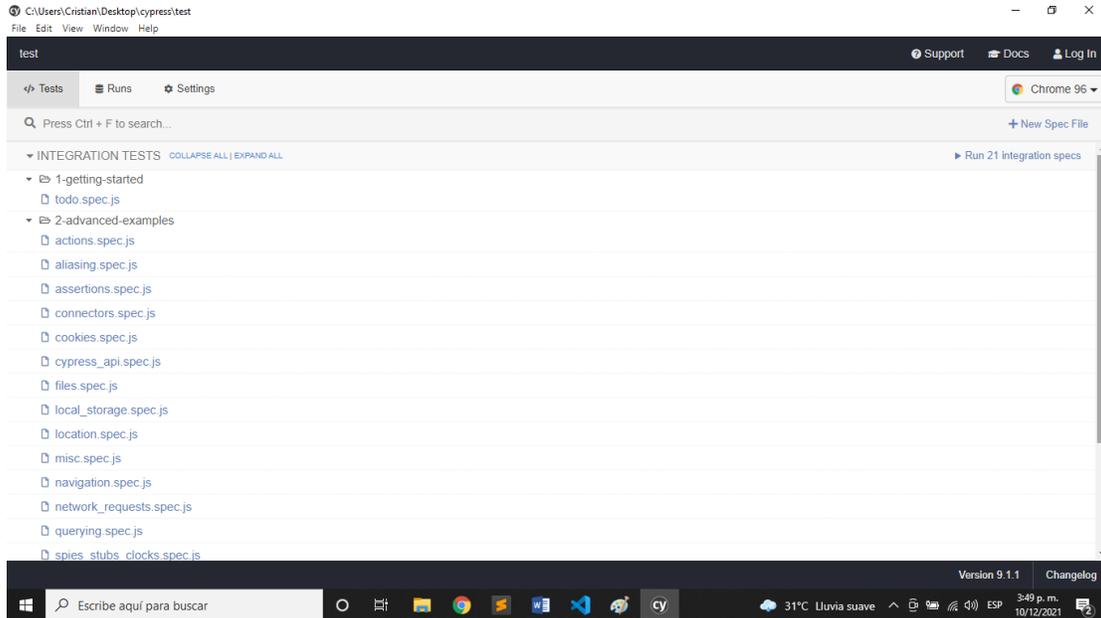


Ilustración 10 Ventana de Cypress

Mediante esta ventana tendremos acceso a las siguientes funciones:

- Seleccionar las pruebas a ejecutar
- Cambiar el navegador donde se ejecutarán las pruebas
- Ver la configuración actual del proyecto
- Crear nuevas pruebas

Para ejecutar nuestras pruebas basta con dar clic sobre el nombre del archivo que queremos ejecutar o en la opción Run se ejecutaran todas las pruebas que se encuentren en la carpeta Integration.

Al momento de iniciar las pruebas se abrirá el navegador que hayamos seleccionado en una ventana como se muestra a continuación.



Informe de Prácticas Profesionales como Opción de Grado

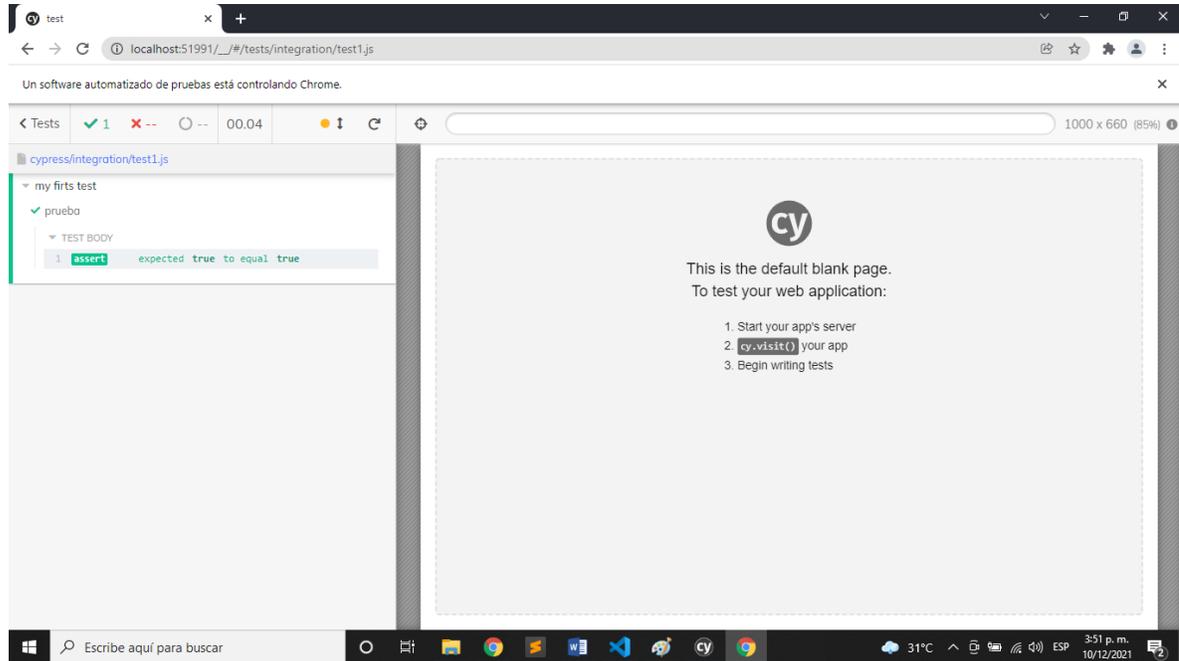


Ilustración 11 Simulación de ejecución de una prueba realizada exitosamente con el framework Cypress

Dentro de esta ventana podremos encontrar los siguientes elementos:

- Ventana del navegador que mostrara como se van ejecutando las pruebas paso a paso
- Nombre de las pruebas y las subpruebas.
- Debug de las subpruebas para poder seguir la ejecución y que se realizó durante cada una de ellas. También podremos observar si cada una de las afirmaciones funciono satisfactoriamente y el resultado final de las pruebas y subpruebas.
- Herramientas de Cypress para reproducir de nuevo las pruebas, pausar las pruebas, opción de selector de elementos del navegador, tiempo de ejecución transcurrido en las pruebas.

Todo lo mostrado en este pequeño ejemplo se realizó con cada una de las pruebas que se estableció en el plan de pruebas automatizadas.



4. EJECUCION PLAN DE PRUEBAS

Se ejecutan los scripts previamente programados para que el sistema realice la simulación del proceso de las pruebas en el aplicativo de forma automática.

```
310 })
311
312 it('login39', function(){
313   cy.get('[placeholder="Correo Electrónico"]').type('circulares@halliburton.com')
314   cy.get('[placeholder="Contraseña"]').type('123')
315   cy.get('.login-buttons > .btn').click()
316   cy.contains('Planeación').click()
317   cy.contains('Postulación a Viajes').click()
318 })
319
320 it('login40', function(){
321   cy.get('[placeholder="Correo Electrónico"]').type('transjoanka@halliburton.com')
322   cy.get('[placeholder="Contraseña"]').type('123')
323   cy.get('.login-buttons > .btn').click()
324   cy.contains('Planeación').click()
325   cy.contains('Postulación a Viajes').click()
326 })
327
328 it('login41', function(){
329   cy.get('[placeholder="Correo Electrónico"]').type('servicumaribo@halliburton.com')
330   cy.get('[placeholder="Contraseña"]').type('123')
331   cy.get('.login-buttons > .btn').click()
332   cy.contains('Planeación').click()
333   cy.contains('Postulación a Viajes').click()
334 })
```

Ilustración 12 Codificación de los scripts en Visual Studio Code.

Acontinuacion se explicaran los resultados del plan de pruebas realizados a la solución seleccionada, para comprobar su correcto funcionamiento y uso.

5. RESULTADO Y REPORTE DE PRUEBAS

Para el reporte de pruebas con el que se validara el correcto funcionamiento de la aplicación, así como el correcto funcionamiento de las pruebas automatizadas para así validar la utilidad de estas, se utilizó una herramienta suministrada por Cypress para este fin, genera automáticamente un reporte de pruebas bastante amplio donde se muestra en tiempo real la ejecución de las pruebas una vez se envíen mediante la consola ejecutando un comando que vinculara las pruebas con la herramienta.



Informe de Prácticas Profesionales como Opción de Grado

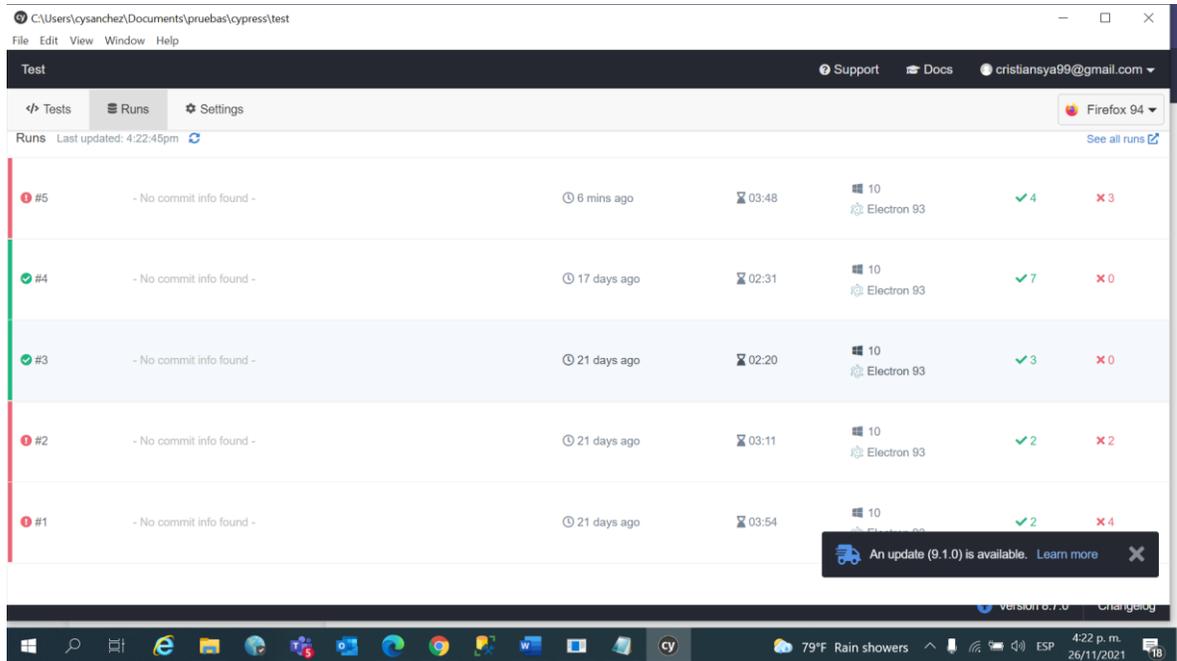


Ilustración 13 Historial de pruebas realizadas vista en el entorno del framework Cypress

Para revisar de manera más profunda el reporte de pruebas Cypress mostrará el resultado de cada prueba de manera individual, con el tiempo que tomo en realizar la prueba y con un video por cada prueba donde se podrá evidenciar de manera más clara el paso a paso de cada prueba para el equipo de desarrollo.



Informe de Prácticas Profesionales como Opción de Grado

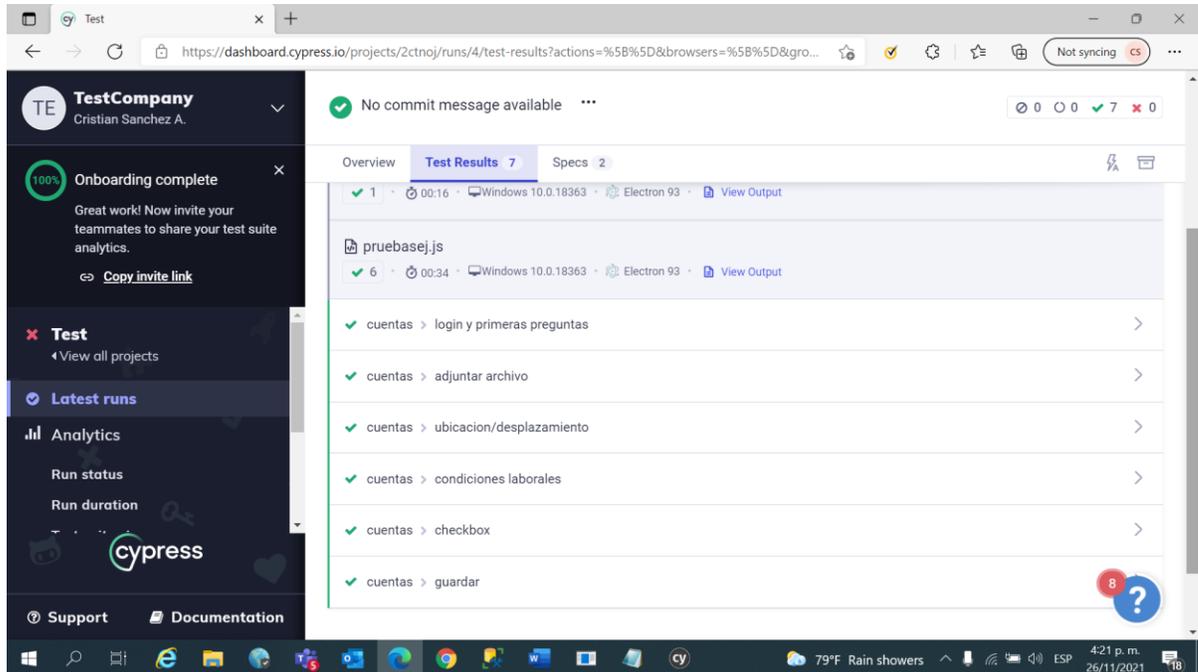


Ilustración 14 Pantalla de visualización de resultado de las pruebas en cypress

Acá se puede observar más a detalle el resultado de los scripts de forma individual de las pruebas automatizadas, donde para este ejercicio podemos observar que todas se realizaron de manera satisfactoria.

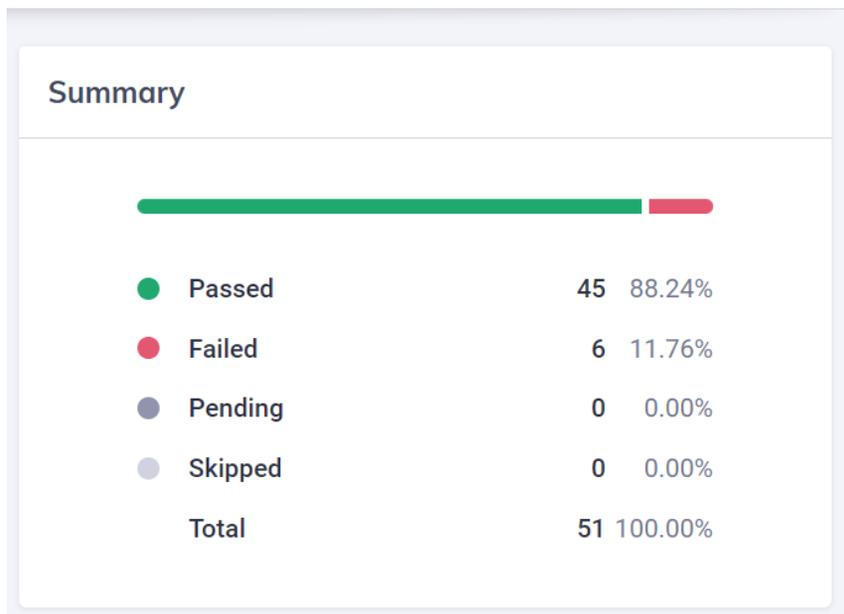


Ilustración 15 Descripción general de las pruebas realizadas



Informe de Prácticas Profesionales como Opción de Grado



En algunos desarrollos encontramos que los casos de prueba no fueron exitosos por lo cual es indispensable reportárselo al desarrollador para su corrección.

```
(Results)
Tests:      51
Passing:    45
Failing:    6
Pending:    0
Skipped:    0
Screenshots: 6
Video:      true
Duration:   45 minutes, 23 seconds
Estimated:  11 minutes, 10 seconds
Spec Ran:   accesohalliburton.js

Spec                Tests  Passing  Failing  Pending  Skipped
x accesohalliburton.js  45:23   51       45       6        -        -
x 1 of 1 failed (100%)  45:23   51       45       6        -        -

(Result)
Tests:      6
Passing:    6
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      true
Duration:   5 minutes, 6 seconds
Spec Ran:   validarcontraseñasfallidas.js

Spec                Tests  Passing  Failing  Pending  Skipped
✓ validarcontraseñasfallidas.js  05:06   6        6        -        -        -
✓ All specs passed!           05:06   6        6        -        -        -
```

Ilustración 16 Visualización de resultados de las pruebas en la terminal de Visual Studio Code

Con este reporte podemos evidenciar la diferencia que implica realizar las pruebas de manera manual contra realizarlas de manera automatizada, un total de 51 casos de prueba que se dividían en subpruebas se realizó en un total de 50 minutos y 29 segundos una gran diferencia a lo que tomaría de manera manual que podría llegar a ser de entre 1 hora 30 minutos y 2 horas para esos escenarios y pruebas, que en



Informe de Prácticas Profesionales como Opción de Grado



caso de presentarse algún error va ser necesarios realizarlos de manera constante lo que aumentara aún más el tiempo asignado para las pruebas, a diferencia de las pruebas que se realizaron de manera automatizada que con un solo comando enviado por consola volverá a ejecutar las pruebas de manera automática sin necesidad de una persona monitoreándolas, solo una revisión al final en el reporte de pruebas. Esto demuestra la gran utilidad y el impacto que implica contar con pruebas automatizadas para un equipo de desarrollo, ahorrando así tiempo y recursos que pueden ser asignados a otras tareas durante el sprint garantizando una mayor calidad en los desarrollos.



Informe de Prácticas Profesionales como Opción de Grado



8. CRONOGRAMA:

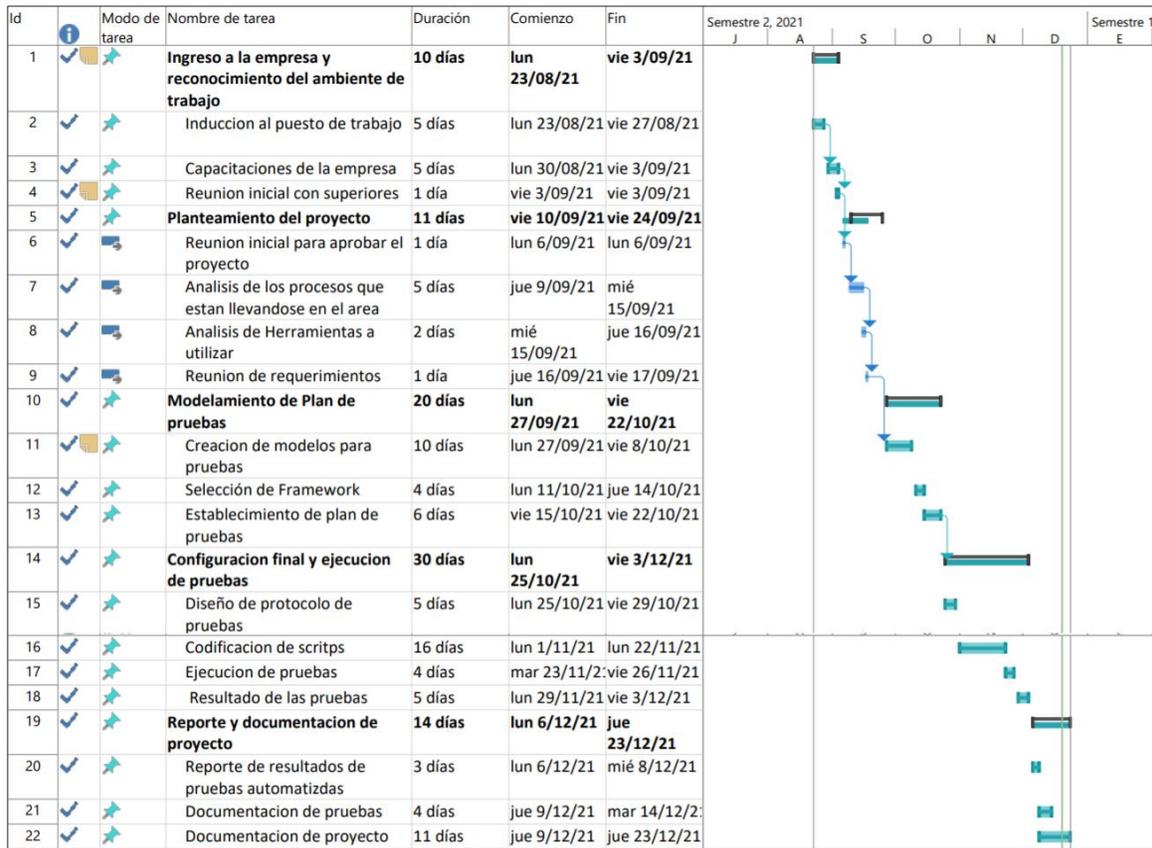


Ilustración 17 Cronograma general de actividades



9. CONCLUSIONES Y LÍNEAS FUTURAS

Contar con pruebas automatizadas es una ventaja competitiva y que garantiza una calidad superior en los productos que se desarrollan, manteniendo actualizado en cada momento al equipo del estado del producto de software, ahorrando recursos, tiempo y personal que deberían apoyar las pruebas manualmente a lo largo de cada uno de los sprints.

El framework Cypress cuenta con todas las herramientas y el ambiente para crear pruebas automatizadas de manera sencilla y en un menor tiempo además al no contar con programas o drivers de terceros garantiza aún más que las pruebas se realizaran en un entorno similar al que el usuario llegaría a interactuar a diferencia de otros frameworks de automatización en el mercado. Aunque realizar las pruebas automatizadas también implica un esfuerzo a largo plazo será totalmente rentable.

Además, es preciso decir que es posible intentar replicar e inclusive migrar a otro Framework en un futuro como Selenium o Playwright puesto que con estos se puede hacer exactamente lo mismo que se hace con Cypress aunque se debe tener en cuenta que ni Selenium ni Playwright tienen un entorno que se adapte tan perfectamente bien a las necesidades de Agile DevOps.

Para finalizar puedo decir que mi experiencia realizando las prácticas en esta empresa ha sido más que enriquecedora y gratificante porque era mi primera experiencia laboral y me he encontrado con un ambiente muy acogedor, personas con una calidad humana muy buena que me guiaron de forma servicial y amigable durante mi proceso de los cuales todos los días aprendía cosas nuevas y siempre conté con su colaboración, además que realizar este proyecto fue todo un reto para mí porque hasta antes de involucrarme en él nunca había trabajado realizando pruebas en aplicativos de software por lo cual era un tema totalmente nuevo para mí y me ha llenado de mucha satisfacción haberlo sacado adelante.



Informe de Prácticas Profesionales como Opción de Grado



10. BIBLIOGRAFÍA

- (2020). *Documentación de Azure DevOps*. Microsoft Docs. <https://docs.microsoft.com/es-es/azure/devops/?view=azure-devops>
- *Documentación framework Cypress*. (2021). Cypress end to end framework. <https://docs.cypress.io/guides/overview/why-cypress>
- *Documentación NodeJS*. (2021). Node.js. <https://nodejs.org/es/>
- Herrera, C. L. (2021, 8 septiembre). *Qué es el Testing de Software*. OpenWebinars.net. <https://openwebinars.net/blog/que-es-el-testing-de-software/>
- *JavaScript | MDN*. (2021, 20 diciembre). JavaScript MDN. <https://developer.mozilla.org/es/docs/Web/JavaScript>
- KPMG International. (2013). KPMG Colombia. <https://home.kpmg/co/es/home.html>
- Ruiz, E. E. (2021, 10 septiembre). *La importancia del QA en el desarrollo de software*. Profile Software Services. <https://profile.es/blog/qa-en-el-desarrollo-de-software/>
- Sebastián Muñoz Tovar. (2021, julio). *AUTOMATIZACIÓN DE PRUEBAS PARA LAS SOLUCIONES DE SOFTWARE AGILE DEVOPS*.