

Informe final de pasantía en la empresa Arroyo Consulting

Laura Vanessa Hernández Benítez

Universidad Nacional Abierta Y A Distancia - UNAD
Escuela De Ciencias Básicas, Tecnología E Ingeniería - ECBTI
Programa De Ingeniería De Sistemas
Diciembre 2021

Agradecimientos

Quiero agradecer principalmente a Dios por permitirme estar a un paso de culminar mi carrera como profesional después de muchos años de estudios y de intentos anteriores, agradecer principalmente la vida y las bendiciones que Dios nos da día a día.

Gracias a mis padres y familiares cercanos que me han apoyado en este proceso, que han creído en mí y que me han motivado en los momentos difíciles a seguir con este camino de aprendizajes; agradezco especialmente a mi esposo Juan Carlos y mi hija recién nacida Celeste por apoyarme y darme el tiempo de terminar mis estudios, gracias a mi esposo por ser tan comprensivo he incondicional por ayudarme en otra labores para poder dedicar tiempo a mis estudios, a mi madre porque la llegada de mi bebé ha sido más llevadera con su ayuda y apoyo, gracias a mi hermanita porque también nos ha ayudado en este proceso.

También a la universidad por los conocimientos brindados, por la facilidad para finalizar mis estudios remotamente y gracias a los docentes que me han apoyado en este camino de estudios y adquisición de conocimiento. Gracias a la metodología que maneja la universidad he podido culminar mis estudios de manera remota y he tenido la libertad de vivir en diferentes lugares del mundo sin abandonar mis estudios.

A la empresa Arroyo Consulting por confiar en mis conocimientos desde el inicio, por permitirme seguir creciendo profesional y personalmente, por los compañeros que me han ayudado en este proceso y a los que he tenido la oportunidad de ser su mentora ya que de esta manera es como más he aprendido y adquirido diferentes

habilidades que día a día hacen de mí una mejor profesional. Y agradezco a John Alzate porque desde que tengo la fortuna de conocerlo he crecido mucho profesionalmente, porque él ha creído y confiado en mi desde el inicio y me ha motivado a seguir aprendiendo y lo más bonito de todo es que me ha motivado a compartir mis conocimientos con mis compañeros. ¡Gracias!

Resumen

Este documento detalla la realización de la pasantía realizada en la empresa Arroyo Consulting LLC., en la dependencia de Quality Assurance (Aseguramiento de calidad), la cual se logró con la la realización de diversas soluciones de automatización de pruebas para garantizar la calidad de software desarrollados para un cliente particular. Dichas soluciones fueron diseñadas y construidas con la herramienta llamada Tosca Commander, la cual brinda diferentes opciones para generar y ejecutar casos de pruebas puntuales, en este documento se presenta un breve manual de uso de la herramienta, así como también los resultados obtenidos luego de diseñar y construir los test cases (casos de prueba) automatizados para el cliente; se plantean los objetivos alcanzados en el desarrollo de la pasantía y además se da una percepción de la experiencia vivida por el estudiante durante el ciclo de la misma, con sus logros y dificultades presentadas, así como también la superación de dichas dificultades.

Palabras clave: Automatización de pruebas, Pruebas de software, Tosca Commander

Abstract

This document details the completion of the internship carried out at the company Arroyo Consulting LLC., In the dependency of Quality Assurance, which was achieved with the completion of various test automation solutions to guarantee software quality developed for a particular customer. That solution was designed and built with the Tosca Commander tool, which provides different options to generate and execute specific test cases, in this document a brief user manual of the tool is presented, as well as the results obtained after designing and building automated tests for the client; The objectives achieved in the development of the internship are proposed and, in addition, there is a perception of the experience lived by the student during that, with its achievements and difficulties presented.

Keywords: Automation testing, Quality Assurance in Software, Tosca Commander

Tabla de contenido

Introducción e información general	10
Marco teórico.....	12
1.1 Ciclo de vida de un proyecto de software	12
1.1.1 Análisis y Diseño	12
1.1.2 Construcción / Desarrollo	12
1.1.3 Pruebas.....	13
1.1.4 Automatización de pruebas.....	14
1.1.5 Implementación.....	15
1.2 Metodologías ágiles	17
1.2.1 SAFe.....	17
1.3 Client Experience Portal (CEP).....	18
1.4 DNAV	19
1.5 Soft Skills / Hard Skills.....	19
Contexto de la práctica o pasantía	20
2.1 Razón social.....	20
2.2 Actividad Económica	20
2.3 Reseña Histórica	20
2.4 Misión, Visión y valores.....	21
2.4.1 Visión.....	21
2.4.2 Misión.....	21
2.4.3 Valores.....	21
2.5 Descripción del Organigrama.....	22
Descripción del área en la cual se desarrolla la práctica profesional dirigida o pasantía.....	23
2.6 Organigrama de la Empresa.....	23
2.6.1 Organigrama general de la empresa.....	23

2.6.2	Grupo primario.....	24
2.6.3	Lideres practicos.....	24
2.7	Descripción de la problemática manejada en el desarrollo de la práctica y pasantía	25
	Justificación	27
	Objetivo general.....	28
3.1	Objetivos específicos.....	28
	Desarrollo de las actividades realizadas en la práctica profesional dirigida o pasantía	29
4.1	Cronograma de actividades	29
4.2	Descripción de las actividades realizadas	31
4.3	Desarrollo Plan de actividades	34
4.4	Automatización en Tosca Commander.....	35
4.4.1	Tosca Commander – Manual de uso.....	36
4.4.2	Tosca Commander – Creación de un “Test case”	40
4.4.3	Desarrollo realizado para CEP (Client Experience Portal).....	42
4.4.4	Desarrollo realizado para el proyecto DNAV	44
4.5	Limitaciones confrontadas en la práctica	45
4.6	Aportes del pasante a la Empresa	47
4.7	Productos realizados.....	48
	Anexos.....	50
5.1	Anexo I: Reporte de Ejecución para DNAV	50
5.2	Anexo II: Video de Ejecución en Tosca Commander para DNAV.....	50
5.3	Anexo III: Presentación para Demostración en DNAV	50
	Conclusiones y recomendaciones	51
	Bibliografía.....	52

Lista de Tablas

Tabla 1: Cronograma de actividades (“sprints” de 3 semanas).....	30
Tabla 2: Detalle de Automatización Smoke Suite.....	43
Tabla 3: Detalle de Automatización Regresion Suite	43
Tabla 4: Detalle de Suites para DNAV	45

Lista de ilustraciones

Ilustración 1: Organigrama general Arroyo Consulting. Copyright 2021 por Arroyo Consulting Presentación general.....	23
Ilustración 2: Organigrama - Grupo Primario. Copyright 2021 por Arroyo Consulting Presentación general.....	24
Ilustración 3: Organigrama – Líderes de práctica. Copyright 2021 por Arroyo Consulting Presentación general.....	25
Ilustración 4: Tosca Commander - Módulos, Autoría propia.....	38
Ilustración 5: Tosca Commander -"Test case"s, Autoría propia	39

Introducción e información general

La universidad educa con una serie de conocimientos y habilidades enfocadas en cada área, las cuales son una guía al llegar al mundo laboral. La pasantía es una forma de demostrar los conocimientos y habilidades adquiridas, además que permite vivir el día a día del mundo laboral, y ayuda a abrir los horizontes y adquirir nuevos conocimientos.

Por lo anterior se seleccionó la Pasantía como opción de grado, en este documento se hace un recuento de todo lo vivido y aprendido dentro de los 4 meses de duración de la pasantía, este informe contiene las actividades realizadas y funciones desempeñadas que han favorecido el cumplimiento de los objetivos y resultados planteados a partir de las necesidades de la empresa, así como también demuestra la aplicabilidad de los conocimientos adquiridos durante el ciclo de la carrera de Ingeniería de Sistemas en el área laboral, por la cual se obtiene experiencia útil y además demuestra la adquisición de nuevos conocimientos en el área de trabajo.

Esta pasantía fue realizada en la empresa **Arroyo Consulting LLC.**, en la dependencia de **Quality Assurance** (Aseguramiento de calidad), realizando diversas soluciones de automatización de pruebas para garantizar la calidad de software desarrollados para un cliente particular. Dichas soluciones fueron diseñadas y construidas con la herramienta llamada Tosca Commander, la cual brinda diferentes opciones para generar y ejecutar casos de pruebas puntuales, en este documento se presenta un breve manual de uso de la herramienta, así como también los resultados obtenidos luego de diseñar y construir los test automatizados para el cliente; se

plantean los objetivos alcanzados en el desarrollo de la pasantía y además se da una percepción de la experiencia vivida por el estudiante durante el ciclo de la misma, con sus logros y dificultades presentadas.

Marco teórico

En este capítulo se abordan los conceptos computacionales y metodológicos que se tomaron en cuenta en el desarrollo de esta pasantía. Los conceptos que se tratan a continuación son relacionados con el diseño de software, ingeniería de sistemas, pruebas automatizadas, entre otros términos de software, así como también se realiza una definición inicial de los proyectos en los cuales se desarrolló la pasantía, dicha terminología ayudará a tener un mejor entendimiento de este informe de pasantía.

1.1 Ciclo de vida de un proyecto de software

El ciclo de vida de un software comprende las diferentes fases por las que se debe pasar para la construcción de este, en donde se tiene en cuenta la creación de concepción de la idea del software, el levantamiento de sus requerimientos, diseño, construcción, pruebas, hasta llegar a la implantación y entrega satisfactoria del mismo y su mantenimiento de ser necesario; A continuación se definen las fases y algunos otros términos relacionados con ellas que ayudan a comprender el proceso de pasantía.

1.1.1 Análisis y Diseño

Proceso en el que se define y se decide que funcionalidades debe tener la aplicación, que requerimientos y criterios de aceptación debe cumplir de acuerdo con la definición del cliente o dueño del proyecto.

1.1.2 Construcción / Desarrollo

A partir de las funcionalidades y requerimientos definidos en la fase de diseño, el equipo de desarrollo se encarga de construir el software. En esta etapa se construyen todos los elementos técnicos que necesite el software para cumplir con las

funcionalidades y requerimientos solicitados, ejemplo, se construye la base de datos, se define la arquitectura que tendrá el software, se plantean los lineamientos de desarrollo y artefactos que ayuda a su construcción.

1.1.3 Pruebas

En esta fase del proyecto se debe validar la calidad del software, así como también se garantiza que las funcionalidades y requerimientos definidos en la fase de diseño sean cubiertos a cabalidad en el software desarrollado. Para ello se utilizan diferentes herramientas, usualmente esta fase de pruebas se realiza manualmente.

1.1.3.1 Casos de prueba (Test Cases)

En un conjunto de pasos secuenciales que se debe seguir para probar una determinada funcionalidad, un caso de prueba debe tener los pasos y secuencias, adicionalmente debe tener los criterios de aceptación para que dicha prueba sea válida.

1.1.3.2 Suite de pruebas

Es un conjunto de casos de prueba, dicha suite debe cumplir con un objetivo específico, ejemplo, suite de Smoke, suite de regresión, suite de pagos, etc. Esto dependerá del negocio y de la finalidad que los testers deseen darle

1.1.3.3 Tester

Persona encargada de realizar las pruebas, existen testers manuales que validan la aplicación manualmente y automatizadores, que se encargan de crear pruebas automatizadas para minimizar el tiempo de ejecución manual.

1.1.4 Automatización de pruebas

Esta fase no siempre se incluye en todos los proyectos de software, es más bien una parte adicional de la fase de pruebas, la cual requiere de mayor esfuerzo y dinero que el proceso de pruebas manual, pero ayuda a garantizar la calidad del software en menor tiempo. En la automatización de pruebas se realiza una identificación de los Casos de prueba que se pueden automatizar y a partir de ahí se procede con la construcción de scripts o pruebas automatizadas para minimizar el tiempo de ejecución de dichas pruebas.

1.1.4.1 Automatización de pruebas E2E

Es un tipo de prueba, el cual pretende realizar una validación de punto a punto, es decir, pretende probar una funcionalidad desde el inicio hasta el final.

1.1.4.2 Automatización de pruebas API

API: Un Application Programming Interface (Interfaz de programación de aplicaciones), abreviado como API es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. (Van Lancker, L. 2013).

Las pruebas de API validan que cada API utilizada en la aplicación cumpla con los requerimientos para los que fue construida, así como también valida que la información recibida y retornada por dicha API sea correcta y cumpla con su objetivo.

1.1.4.3 Tosca Commander

Es una herramienta paga que permite la construcción y creación de pruebas automatizadas de diferentes tipos, esta herramienta es de tipo “Record and Play” o grabar y reproducir, la cual permite de una manera muy intuitiva la construcción de “Test cases” automatizados.

El cliente principal de la compañía y para el cual se trabajó en la pasantía tiene adoptada esta herramienta para todo el manejo de Automatización de pruebas, por este motivo se utiliza dicha herramienta en el proceso de la pasantía.

1.1.5 Implementación

Luego de tener el software desarrollado y probado, existe una fase de Implantación, en la cual se despliega el software para que pueda ser accedido, visto y validado por el cliente.

1.1.5.1 Integración Continua / Despliegue continuo (CI/CD)

En la actualidad existen dos términos relacionados esta implantación los cuales son “Integración continua” y “Despliegue continuo” más conocidos como CI/CD.

En la integración continua, se automatiza la integración del código de los desarrolladores luego de pasar por ciertas pruebas sobre dicho código. Permite de diferentes desarrolladores trabajen sobre la base del código y se pueda integrar entre los diferentes cambios realizados.

En el Despliegue continuo, se automatiza la liberación de versiones del software o despliegues del software probado y validado más para que sea más continuamente validado y aprobado por el cliente en menos tiempo, y así identificar posibles errores o cambio necesarios en el software.

1.1.5.2 Ambientes

Un ambiente es un espacio en un servidor en donde se despliega una versión específica de la aplicación para su ejecución y validaciones, existen ambientes bajos, son los usados por los desarrolladores y testers para construir y probar el software, usualmente existen ambientes de desarrollo y pruebas separados ambiente de “DEV” y ambiente de “QA” respectivamente; y ambientes altos, que son usados por los dueños del software o clientes para realizar sus validaciones antes de publicar una versión de este al público objetivo, usualmente se usa ambiente de “STAGE” y “LOAD”.

1.1.1.1 Azure DevOps

Es una aplicación de Microsoft que permite el manejo de los proyectos en todo su ciclo de vida, este permite desde el manejo de la planeación, análisis y diseño del proyecto, construcción de “test cases” y “suites de pruebas”, manejo de los Repositorios de código, pipelines y despliegues.

Esta es la herramienta aprobada por el cliente y la que se utilizó en el transcurso de toda la pasantía para la gestión de las tareas, asignaciones, “test cases” y “suites” manejadas para los diferentes proyectos.

1.1.5.3 Pipeline

Es una secuencia de pasos que se definen para automatizar una tarea, esencialmente se usa para automatizar las tareas de automatización de CI/CD.

1.2 Metodologías ágiles

Las metodologías ágiles buscan reducir el tiempo en el que se ven entregas del proyecto, realizando todo el proceso comprendido en el ciclo de vida del software, pero en iteraciones pequeñas, que abarquen pequeñas funcionalidades o hitos que requiere la aplicación e realizando pequeños entregables funcionales que permitan hacer un software de manera iterativa e incremental. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. La metodología ágil favorece un enfoque sencillo de la documentación de software, y acepta los cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de resistirse a ellos (Amaro Calderón, S. D., & Valverde Rebaza, J. C. 2007).

1.2.1 SAFe

(acrónimo de Scaled Agile Framework Enterprise) es un marco de trabajo para la escalación de las prácticas ágiles basado en los principios de Lean y Agile para el desarrollo de software y sistemas, a nivel corporación. (Hepner, L. 2019).

1.2.1.1 Sprint:

Un sprint es un período breve de tiempo fijo en el que un equipo de scrum trabaja para completar una cantidad de trabajo establecida. (Martínez, J. I., & Comino López, M. 2018)

1.2.1.2 Daily Meeting o reunión diaria:

Es una reunión diaria que mantiene al equipo informado, conectado y calibrado a lo largo del sprint, en esta reunión se contestan las preguntas básicas ¿Qué hice ayer? ¿Qué planeo hacer hoy? ¿Qué impedimentos o bloqueos tengo?, estas preguntas deben ser contestadas por cada miembro del equipo para que todos estén informados del estado del sprint y si se va a lograr cumplir los objetivos del sprint definido en el “Planning”.

1.2.1.3 Planning:

La planificación de “Sprint” es un evento o reunión en donde se define qué se puede entregar en el “sprint” próximo y cómo se va a conseguir ese trabajo, esta reunión se realiza antes de empezar el “sprint”.

1.2.1.4 Demo o demostración

Es una reunión donde se presentan a todo el equipo, incluido el cliente las nuevas funcionalidades y cambios desarrollados en el software de acuerdo con la planeación del sprint, el objetivo de esta demostración es mostrar el trabajo realizado durante el sprint.

1.3 Client Experience Portal (CEP)

Aplicación tipo “Dashboard” que permite ver el resultado de 6 de las aplicaciones del cliente, esta se encarga de consumir de otro software la información más importante y mostrarla como tipo tablero de control. Es software que integra los demás softwares del cliente para tener una visión amplia y clara.

1.4 DNAV

Aplicación creada para manejar las auditorias que realiza la empresa del cliente a sus propios clientes, en ella se pueden crear diferentes auditorias, hacer su seguimiento y control y seguir los procesos que la auditoría requiere para ser finalizada. También cuenta con otros

1.5 Soft Skills / Hard Skills

Los “skills” o habilidades son parte fundamental de una persona y define que capacidad tiene desarrolladas. Estos “skills” se dividen en dos, “soft skills” o habilidades blandas y “hard skills” habilidades duras; los “soft skills” son habilidades que como persona se deben desarrollar para tener una mejor relación entre sí, por ejemplo, comunicación asertiva, empatía, trabajo en equipo, compromiso, entre otras. Los “hard skills” son habilidades mas técnicas que se adquieren para desarrollar diferentes trabajos, por ejemplo, un “hard skill” para un desarrollador es que conozca algún lenguaje de programación y que sepa programar.

Contexto de la práctica o pasantía

2.1 Razón social

Empresa **Arroyo Consulting LLC**. Ubicada en Nashville, Tennessee, Estados Unidos. Esta empresa es creada y dirigida por un Ingeniero de sistemas antioqueño, motivado en brindar excelentes servicios de tecnología a empresas de diferentes partes del mundo con personal latino americano.

2.2 Actividad Económica

Es una compañía de clase mundial que provee servicios de tecnología soportado en el mejor talento humano disponible en la región, ayudando a sus clientes a crecer sus organizaciones a través de la implementación de soluciones digitales; la empresa al ser un creador de soluciones digitales para terceros brinda las diferentes etapas comprendidas en el ciclo de vida del software a sus clientes, creando así una solución integral diseñada y ajustada a cada cliente y a cada requerimiento.

2.3 Reseña Histórica

Arroyo Consulting es una empresa de servicios tecnología de clase mundial, su experiencia en desarrollo de software con metodologías ágiles los ha llevado a proveer software y servicios de TI a grandes compañías de clase mundial por más de 13 años (Deloitte, Brinks, Bridgestone, FedEx), esta empresa puede ayudar con el desarrollo de software en diferentes tecnologías, los servicios en América latina y norte América tienen precios competitivos y por las locaciones comparte la misma zona horaria con estados unidos.

2.4 Misión, Visión y valores

2.4.1 Visión

En el 2023 desean ser líderes en el sector de TI implementando soluciones en tecnología desde la región al mercado norteamericano, enmarcado en el crecimiento de la compañía a través del desarrollo humano y profesional de equipos sólidos que los conviertan en aliados estratégicos para sus clientes por esforzarse en superar las expectativas de adaptación, desempeño y calidad de sus clientes.

2.4.2 Misión

Ser una empresa de clase mundial en servicios de tecnología que ponga a disposición de sus clientes lo mejor del talento humano para ayudarlos en el crecimiento sus organizaciones a través de la implementación de soluciones digitales. Por medio de esto, traer oportunidades que tengan un impacto positivo en la región.

2.4.3 Valores

Adaptación: Que permita responder ágilmente a las necesidades de sus clientes de forma eficaz con rápidos ciclos de aprendizaje.

Compromiso con la Excelencia: Que mantenga viva la llama de la pasión por lo que hace la compañía en una incesante búsqueda del mejoramiento con los recursos disponibles.

Autogestión colaborativa: Que permita trabajar hacia resultados y entrega de valor siendo parte de un colectivo en constante sincronización y apoyo.

2.5 Descripción del Organigrama

Arroyo sabe que sus desarrollos son creados por humanos y para humanos, por esta razón considera a sus colaboradores como su activo más valioso. Por eso los cuida y les ofrece herramientas de aprendizaje, capacitaciones en “soft” / “hard” “skills”, flexibilidad en sus horarios entre otros beneficios para que ellos se sientan trabajando como en su hogar.

Por lo anterior la compañía Arroyo Consulting tiene una distribución de organigrama entre horizontal y circular, ya que la empresa no tiene jerarquías definidas; la empresa está conformada por un grupo primario y un grupo de líderes o facilitadores por área que se encargan apoyar a los demás colaboradores de ser necesario, y fomentan el aprendizaje por medio de capacitaciones y asesorías.

Descripción del área en la cual se desarrolla la práctica profesional dirigida o pasantía

La pasantía se desarrolló en el área de la Automatización de Software, la cual hace parte del gran proceso de Aseguramiento de la Calidad del software y es parte fundamental del ciclo de vida de este.

2.6 Organigrama de la Empresa

2.6.1 Organigrama general de la empresa

En la Ilustración 1 se muestra el organigrama general de la empresa Arroyo Consulting, donde se identifican los dos principales grupos de liderazgo, el grupo primario y los líderes de prácticas

Ilustración 1

Organigrama general Arroyo Consulting. Copyright 2021 por Arroyo Consulting
Presentación general



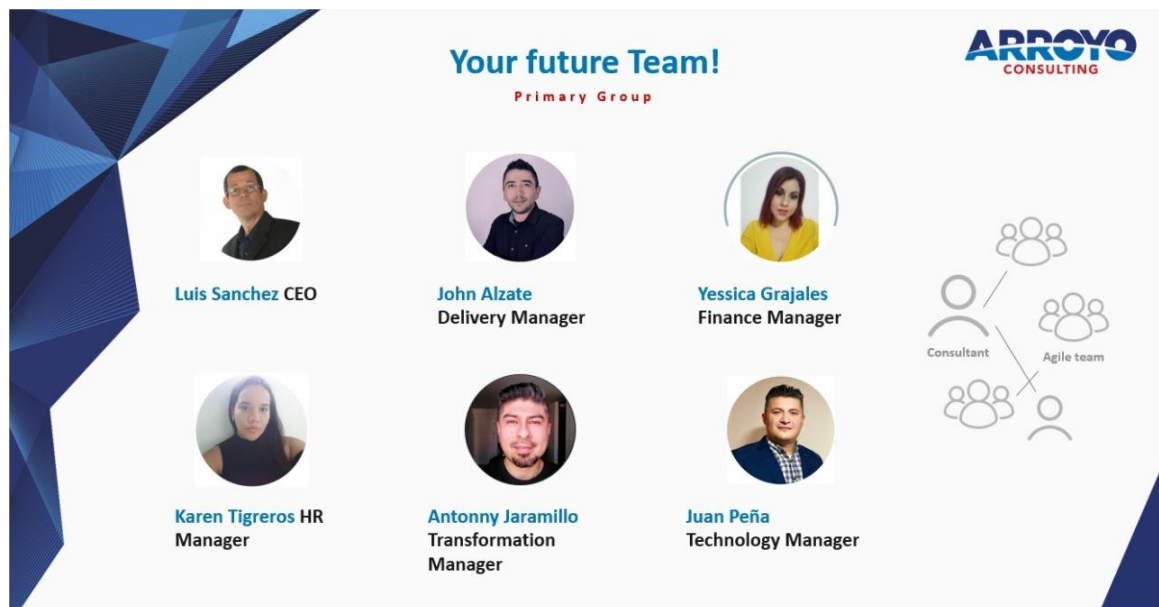
2.6.2 Grupo primario

En la Ilustración 2 se muestra el detalle de los participantes del grupo primario, ellos se encargan de que a compañía este alineada, que todo fluya para brindar el mejor servicio a sus clientes,

Ilustración 2

Organigrama - Grupo Primario. Copyright 2021 por Arroyo Consulting

Presentación general



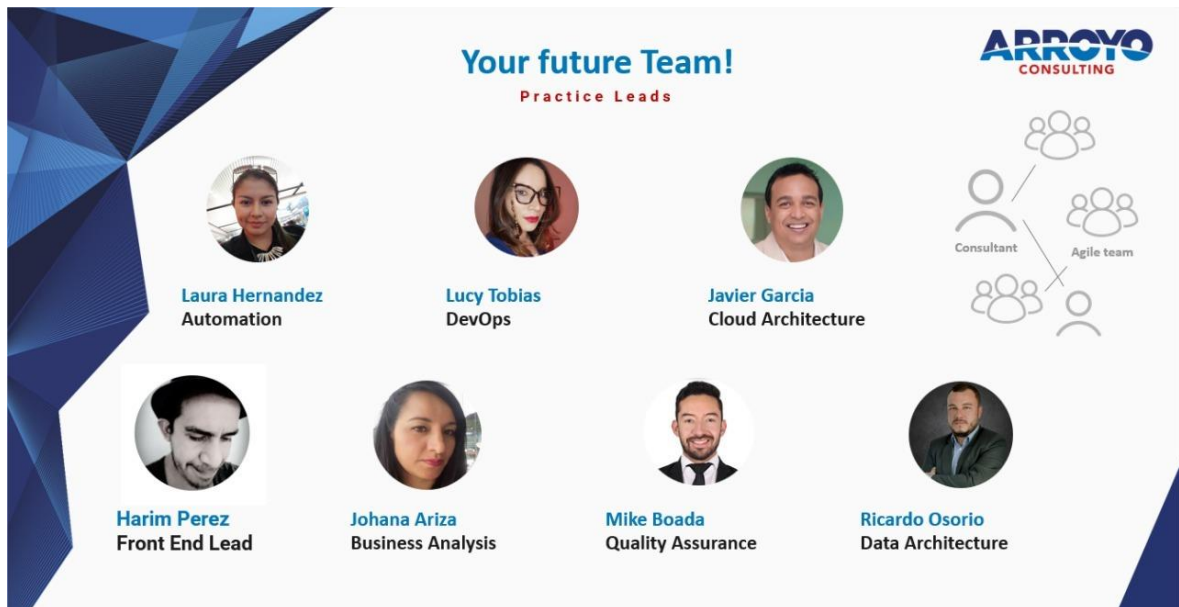
2.6.3 Líderes prácticos

En la Ilustración 3 se muestra el detalle de los líderes técnicos de la compañía. Los cuales se encargan de la formación y crecimiento profesional y técnica de las personas de cada una de sus áreas, brindan cursos, asesoría, y sobre todo motivación de crecimiento y mejora continua.

Ilustración 3

Organigrama – Líderes de práctica. Copyright 2021 por Arroyo Consulting

Presentación general



2.7 Descripción de la problemática manejada en el desarrollo de la práctica y pasantía

El proceso de Pruebas de software es una de las fases que debe llevarse a cabo para garantizar que dicho software cumpla con la calidad necesaria para ser entregado. Cuando este proceso es netamente manual (Pruebas Manuales) se llevan a cabo tareas repetitivas, se vuelve un trabajo dispendioso, costoso y complejo, y más aún cuando el software que se está probando tiene diferentes integraciones, procesos, versiones, etc.

Por este motivo es necesario la implementación de las pruebas automatizadas, ya que al crear scripts que se ejecuten periódicamente se reduce significativamente el tiempo en pruebas manuales, así como también se mitigan más errores, se puede

cubrir una mayor cantidad de pruebas en menos tiempo y se garantiza de una mejor manera la calidad del software y de los componentes que este tiene.

Para el proyecto “Client Experience Portal” y “DNAV” se identificó la necesidad de creación de pruebas automatizadas, ya que ambos softwares tienen múltiples integraciones con otros sistemas, y además cierta complejidad a la hora de hacer pruebas de regresión y toma mucho tiempo manualmente para ello. Por este motivo se trata de reducir los tiempos de ejecución de pruebas de regresión y pruebas de Smoke test con la creación de pruebas automatizadas y agilizar la entrega y despliegue de nuevas versiones del software al cliente.

Justificación

El proyecto Client Experience Portal es un software que sirve como Dashboard para unificar la información de muchas otras aplicaciones del cliente, por este motivo tiene diferentes integraciones y recibe información de varias aplicaciones que brindan los datos principales que en este software se presentan.

Al este proyecto tener tantas integraciones es necesario garantizar su calidad y validar que dichas integraciones sean correctas, que traigan adecuadamente a la información de cada uno de los sistemas que usan y los datos correctos. Esta validación se realiza manualmente, pero también se debe realizar automáticamente para que cada se realice un cambio en mismo sea más rápido y fácil asegurar la calidad de las funcionalidades del sistema y sus diferentes integraciones.

Objetivo general

Asegurar la calidad del proyecto Client Experience Portal con la generación y ejecución de pruebas automatizadas poniendo en marcha un plan de pruebas definido para este, apoyando el equipo de pruebas manuales y equipo de desarrollo con la identificación temprana de bugs y asegurando el correcto funcionamiento de la aplicación luego de cada despliegue o entrega del proyecto, esto por medio de la ejecución automática de las pruebas desde la integración continua del mismo.

3.1 Objetivos específicos

Diseñar y definir un plan de pruebas donde se incluya automatización de acuerdo con los requerimientos propios del proyecto, identificando los escenarios de prueba que se deben automatizar y cuáles no.

Estructurar la arquitectura base del proyecto de pruebas de acuerdo con el plan de pruebas.

Diseñar y construir los diferentes scripts de prueba para automatizar los escenarios de prueba definidos.

Construir los scripts de prueba para garantizar la calidad de la parte funcional y visual del sistema (Pruebas de E2E).

Asegurar la ejecución de pruebas automáticas de acuerdo con el plan de pruebas definido.

Desarrollo de las actividades realizadas en la práctica profesional dirigida o pasantía

4.1 Cronograma de actividades

De acuerdo con las metodologías ágiles utilizadas por la empresa cliente, se utiliza la metodología SAFe para todos los equipos de trabajo, dentro de este marco de trabajo también se utiliza el desarrollo por “sprints”, En cada “Sprint” o cada ciclo de trabajo a conseguir es lo que se denomina un entregable o incremento del producto, que aporte valor al cliente”. (Requena Mesa, Abraham, 19 de diciembre de 2018).

La siguiente tabla contiene el cronograma de trabajo planeado por sprints de 3 semanas, donde se presenta la actividad a realizar y en que sprint se debe realizar

Tabla 1:*Cronograma de actividades (“sprints” de 3 semanas)*

ACTIVIDAD	Sprint 0	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Transferencia de conocimiento	X	X				
Construcción de Plan de pruebas	X	X				
Pruebas Manuales	X	X	X			
Crear la ruta crítica de la aplicación	X					
Construir la arquitectura base del Proyecto	X	X				
Generar métricas de ejecución de pruebas Manuales y Automatizadas	X	X	X	X	X	X
Reporte de bugs detectados	X	X	X	X	X	X
Integración de pruebas automatizadas en CI/CD Pipeline			X	X		
Diseñar los casos de prueba de UI que se deben automatizar		X	X			
Construcción de pruebas automatizadas de E2E		X	X	X	X	X
Generar reporte de ejecución de pruebas automáticas				X	X	X
Presentación demostrativa (DEMO) de pruebas automatizadas			X	X	X	X
Mantenimiento de pruebas				X	X	X

4.2 Descripción de las actividades realizadas

Entre las actividades desarrolladas en el ciclo de la pasantía, se encuentran:

- **Creación de Plan de pruebas**

Proceso que se realizó en la primera etapa donde se detalló cómo se iba a abordar el proceso de automatización de pruebas, esta planeación se realizó con el apoyo de todo el equipo de QA y BSA para determinar que módulos/requerimientos/"Test cases" se deberían tener en cuenta para la automatización.

- **Creación de tareas a realizar**

Como se mencionó anteriormente el plan de trabajo se debe realizar por "sprints", lo que también afecta la creación y asignación de tareas; por medio de la herramienta *Azure DevOps* se realizó el control de las tareas a realizar y de las tareas pendientes, así como también los "Test cases" automatizados y los que no se deben automatizar.

- **Selección de casos de prueba a automatizar**

Para esta selección de casos de prueba a automatizar, se debe tener en cuenta que los desarrollos (Partes de la aplicación a probar con pruebas automatizadas) deben estar completos y testeados manualmente. Por esta razón se trabajó en paralelo con la planeación de cada sprint anterior y de acuerdo con ello se fueron seleccionando los casos de prueba que iban incluyendo en la automatización.

- **Automatización de las pruebas seleccionadas en la herramienta Tosca Commander**

De acuerdo con las tareas creadas en cada sprint y los requerimientos testeados y desarrollados en los sprint anteriores, se realizó la seleccionando y procedía con la construcción de los scripts para automatizar cada uno de los “Test cases” incluidos en las tareas. Las pruebas automatizadas deben cumplir con los criterios de aceptación del “Test case” al que hace referencia la prueba.

- **Ejecución de pruebas por medio de la herramienta Tosca Commander**

En cada sprint se realizaba una ejecución manual (Ejecución desde la herramienta Tosca Commander en ambientes locales) de los “Test cases” automatizados previamente, esta ejecución nos permite saber si las funcionalidades creadas previamente seguían teniendo el mismo comportamiento y la buena calidad con la que se crearon, o si por el contrario había algún error por corregir.

- **Ejecución de pruebas desde el pipeline**

Esta ejecución es un poco diferente, ya que es una ejecución desde una maquina remota y es una ejecución de las pruebas automatizadas al realizarse un despliegue en cierto ambiente (QA, Stage, Load). Luego de dicha ejecución de genera un reporte y se toman acciones dependiendo de los resultados obtenidos.

- **Generación de reportes de ejecución**

Esta generación se hace luego de realizar la ejecución de las pruebas automatizadas y permite ver el estado de las mismas. Este reporte es generado los últimos días de cada Sprint para llevar el control de la calidad de la aplicación.

- **Demostración de las pruebas realizadas**

En cada sprint se realiza una reunión con todo el equipo de trabajo (incluido el cliente) para presentar los “Test cases” completados y automatizados por sprint. En este demo se realiza la explicación de los “Test cases” automatizados, se muestra la ejecución de dicha automatización y adicionalmente se presenta el reporte de ejecución de estos. Esta demostración ayuda a todo el equipo de trabajo, incluido el cliente a que conozca el estado de calidad del software, así como también tenga claro que parte de este está o no incluido en las pruebas automatizadas.

- **Revisión y validación de las automatizaciones**

Así como existe un proceso de creación de los scripts para la automatización, también se incluye un proceso para la revisión y validación para verificar que dichos scripts sean correctos. En esta revisión se valida que el script cumpla a cabalidad con la definición del “Test case” que está cubriendo, así como también se revisa coherencia y cumplimiento de las buenas prácticas de automatización definidas por el equipo de automatización.

- **Mantenimiento y/o corrección de los scripts automatizados**

Como es usual, el software va cambiando de acuerdo con los requerimientos del cliente, y estos cambios son totalmente aceptados al trabajar con metodologías ágiles, por esta razón la automatización de las pruebas también se debe adaptar a dichos cambios. El mantenimiento y de los “Test cases” automatizados es una de las tareas más usuales ya que por diferentes motivos un “Test case” puede fallar y se debe realizar la respectiva validación y revisión de este, así como también su corrección si es del caso.

- **Acompañamiento y asesoría a los nuevos integrantes**

En este caso, al pasante ser persona en la empresa con más experiencia en el uso de la herramienta Tosca Commander, es su deber acompañar y asesorar a los nuevos compañeros que inician a trabajar con dicha herramienta, bien sea en el proyecto asignado o en otros proyectos.

- **Documentación de errores/Creación de Bugs**

Cada vez que en la ejecución de las pruebas se detecta un bug se debe reportar para que este sea solucionado; una vez solucionado y testeado manualmente se debe realizar nuevamente la ejecución de las pruebas automatizadas para validar su correcta ejecución.

4.3 Desarrollo Plan de actividades

Dentro del plan de actividades no se tienen en cuenta actividades relacionadas con inducción o introducción a la empresa, ya que desde octubre del año pasado (2020) el pasante se encuentra laborando con esta gran compañía y adicionalmente tiene un

par de años de experiencia en este rol y en diferentes roles relacionados con Ingeniería de sistemas en empresas en las que ha trabajado anteriormente.

A continuación, se definen las tareas generales que se realizan en cada uno de los “sprints” (de 3 semanas cada uno):

- Reuniones diarias (“Dailies”)
- Reuniones de planeación de sprint
- Reunión de revisión de los casos de prueba a automatizar con el equipo de pruebas manual
- Creación de las actividades específicas por cada sprint
- Reuniones de retroalimentación/retrospectiva de cada sprint
- Reuniones adicionales de negocio de acuerdo con las actividades asignadas
- Creación de Scripts de automatización en Tosca Commander de acuerdo con las actividades asignadas y a los “Test cases” seleccionados para la automatización
- Demo en cada sprint para mostrar las pruebas de automatización finalizadas en el mismo, así como también el reporte de ejecución de estas
- Revisión de los scripts generados por otros compañeros
- Mantenimiento de los Scripts (“Test cases”)

4.4 Automatización en Tosca Commander

A continuación, se presenta un breve manual sobre la herramienta Tosca Commander, donde se explican los elementos claves para la creación de un script para la automatización de un “Test case”. También se presentará la automatización creada para el proyecto CEP (Client Experience Portal).

4.4.1 Tosca Commander – Manual de uso

Tosca commander es una herramienta paga utilizada para la creación de prueba automatizadas tanto para pruebas E2E como para pruebas de API. Esta herramienta es bastante intuitiva y amigable, la mayoría de sus acciones son muy dinámicas y visuales, a diferencia de otras herramientas de automatización no requiere de la creación de scripts a través de algún lenguaje de programación, sino que ella misma facilita la posibilidad de crear los “Test cases” automatizados minimizando la utilización de lenguajes de programación.

El código generado por Tosca Commander queda almacenado como una base de datos, en un archivo con extensión “.bd”.

Empezaremos definiendo los principales tipos de elementos que se pueden crear con esta herramienta y que nos facilitan la creación de los “Test cases” automatizados. La siguiente explicación e imágenes fueron tomadas en el proyecto desarrollado al realizar el curso “Automation Specialist Level 1” brindado por la empresa Tricentis creadora de dicha herramienta.

4.4.1.1 Módulos

Para iniciar la creación de un caso de prueba (“Test case”) en Tosca, primero es necesario realizar la creación de un Módulo, los módulos son los elementos que tienen mapeada la estructura de la página que se va a probar y todos los elementos a probar, es decir la estructura en HTML de la página.

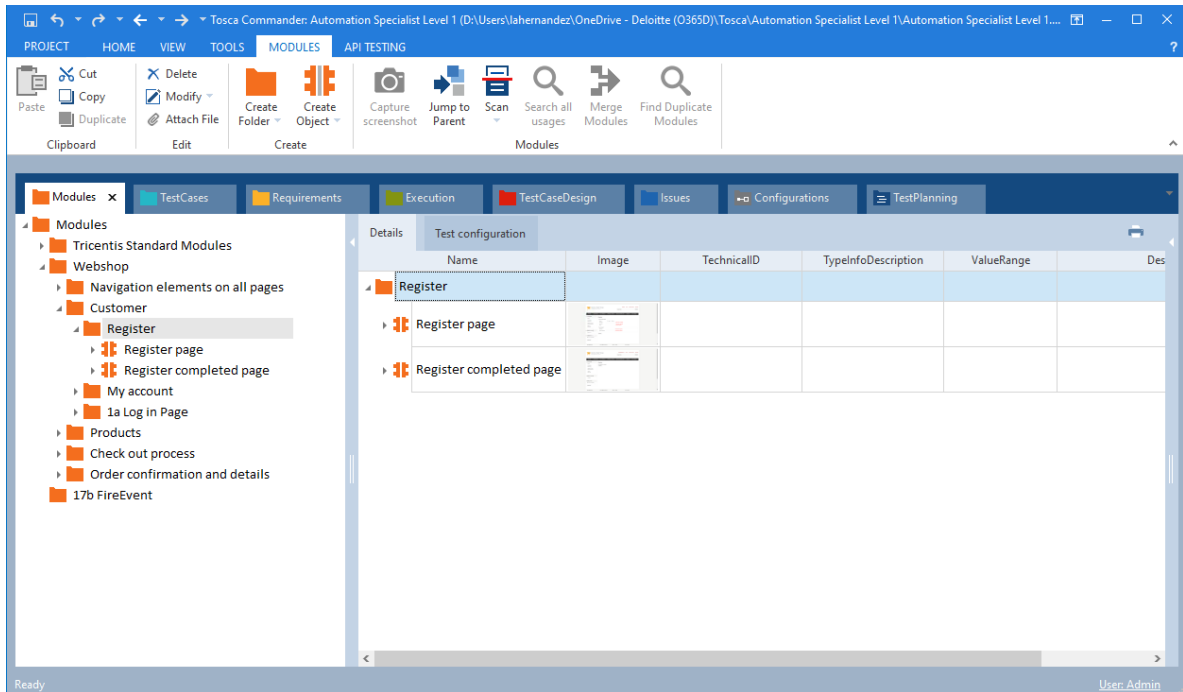
Para crear un módulo se deben seguir los pasos mencionados a continuación:

1. Abrir nuestro navegador
2. Navegar hasta la página que se desea testear dentro de la aplicación específica
3. Desde Tosca Commander en la parte de Módulos se puede realizar el escaneo de dicha página para crear el módulo.
4. Clic derecho en la carpeta de Modulo donde se desea crear el módulo
5. Clic en “Scan” > Application
6. Se abre una ventana emergente, donde se muestran todas aplicaciones que se encuentran abiertas en el pc donde se está ejecutando Tosca
7. Seleccionar el navegador donde se encuentra abierta la aplicación a testear
8. Se abre una ventana emergente, donde se pueden seleccionar los elementos de HTML que se deben
9. Seleccionar los elementos que se van a usar en el “Test case” y los que se puede usar en los demás “Test cases” a probar
10. Guardar el módulo

Luego de escanear nuestra página con la ayuda y facilidad del scanner de Tosca, tendremos nuestro módulo listo para ser usado en un “Test case”. En la ilustración 4 se muestra la opción de Módulos dentro de Tosca Commander

Ilustración 4

Tosca Commander - Módulos, Autoría propia



4.4.1.2 Casos de prueba - “Test cases”

Luego de tener creado el módulo a probar, podemos proceder con la creación del “Test case” y sus respectivos pasos y validación, para ello se pueden seguir los pasos definidos a continuación

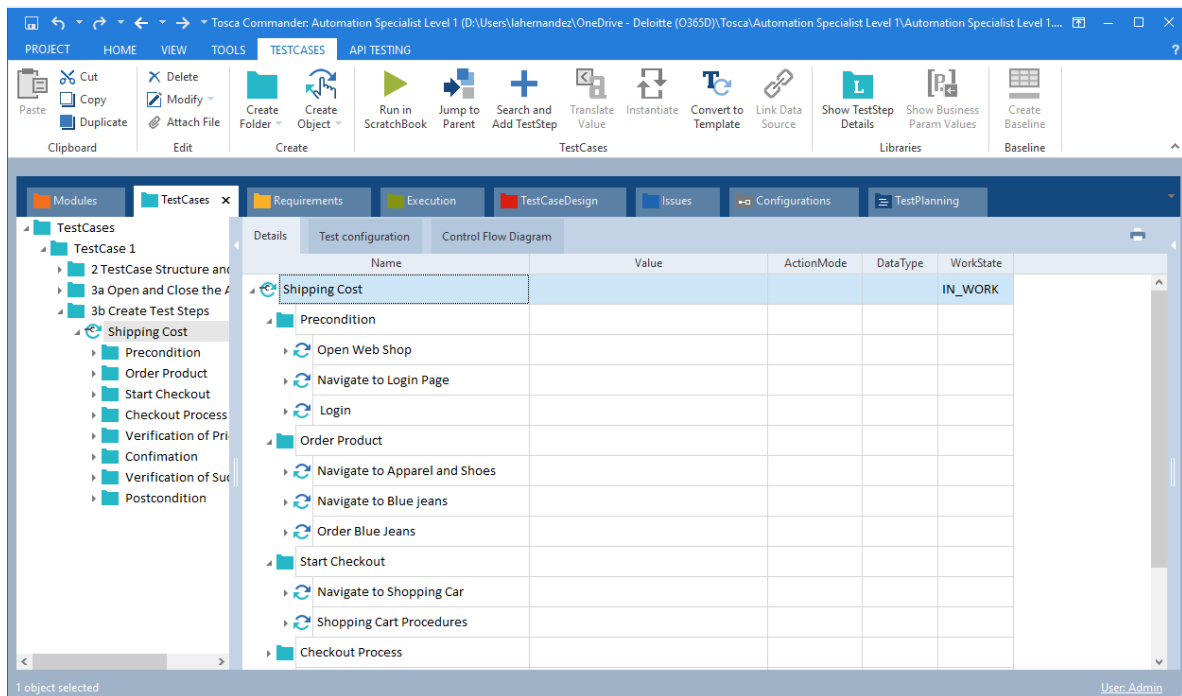
1. En la parte de “Test cases” en Tosca
2. Crear el “Test case” con su respectivo nombre
3. Para crear un paso dentro del mismo se debe crear haciendo “drag and drop” (arrastrar y soltar) del módulo creado anteriormente sobre el nuevo “Test case”, de esta forma se creará el paso

4. Dentro del paso creado se pueden agregar las validaciones, condiciones y demás verificaciones necesarias para cumplir los resultados esperados del “Test case”.

Nota: se pueden crear diferentes pasos con el mismo modulo para separar las acciones de las verificaciones. En la ilustración 5 se muestra la opción de Test case dentro de Tosca Commander

Ilustración 5

Tosca Commander -” Test cases”, Autoría propia



4.4.1.3 Listas de Ejecución

Luego de tener creados los “Test cases”, se pueden crear las listas de ejecución; como su nombre lo indica las listas de ejecución sirven para ejecutar los “Test cases”.

Estas listas son creadas a partir de los “Test cases” o “Test case folders” que se deseen agregar a dicha lista.

Desde estas listas se pueden ejecutar los “Test cases” y obtener un reporte de dicha ejecución, esta ejecución puede ser en el ambiente local (donde está instalado la herramienta Tosca Commander y desde donde se crean los “Test cases” y demás elementos) o desde un Agente de Tosca (Máquina remota que obtiene los últimos cambios del repositorio y los ejecuta)

4.4.2 Tosca Commander – Creación de un “Test case”

A continuación, se presenta un pequeño ejemplo de cómo se realiza la creación de un “Test case”, este será realizado con ayuda de la página de prueba de Tosca.

4.4.2.1 Creación del Módulo.

1. Abrimos el navegador y la página que deseamos probar
2. Realizamos el escaneo de la página con ayuda de la opción “Scan” de Tosca Commander, Clic derecho en la carpeta y Clic en la opción “Scan”, en el submenú se debe seleccionar la opción application
3. Se abre una ventana donde se presentan todas las aplicaciones que se encuentre abiertas en el pc
4. Seleccionamos el navegador donde tenemos nuestra aplicación abierta
5. Se abre la ventana de “Tosca Scanner”, donde podemos ver los elementos de la aplicación, seleccionarlos y configurarlos de acuerdo con las necesidades del “Test case”
6. Seleccionamos los elementos que queremos tener en cuenta para los “Test case”
7. Guardamos el modulo

4.4.2.2 Creación del “Test case”

1. Creamos el “Test case” en la carpeta deseada, con el nombre adecuado
2. Hacemos “drag and drop” del módulo sobre el “Test case” anteriormente creado, esto nos permite crear un paso dentro del “Test case”
3. Agregamos las validaciones, acciones y demás elementos requeridos por el “Test case”, en este caso vamos a hacer varios pasos (copiamos el paso creado anteriormente) para realizar el ingreso de la información, la acción de Clic y la validación, es decir 3 pasos creados con el mismo modulo
4. A continuación, vamos a crear el primer paso con el ingreso de los datos
5. El segundo paso lo ajustamos para que realice el Clic en el botón
6. El tercer paso lo vamos a usar para realizar la validación de la acción realizada anteriormente

4.4.2.3 Ejecución del “Test case”

1. Por medio de la acción “Run in Scrashbook” realizamos la ejecución del “Test case”
2. Vemos como la aplicación abre automáticamente el enlace de la aplicación (precondición agregada)
3. Se ingresan automáticamente los datos
4. Hace Clic en el botón
5. Realiza la verificación
6. Muestra una ventana con los resultados de ejecución, en esta ventana se pueden ver los errores en caso de que el “test case” falle o el resultado positivo

4.4.2.4 Creación del Execution List

1. Creamos la lista de Ejecución en la carpeta deseada, con el nombre de la lista que deseemos poner
2. Hacemos “drag and drop” de la carpeta donde se encuentra el o los “Test cases” que deseamos agregar a la lista de ejecución
3. Queda la relación de los “Test cases” con la lista de ejecución

4.4.2.5 Ejecución de la Execution List

1. Clic derecho sobre la lista de ejecución
2. Usando la opción Run se corre la lista de ejecución y se ejecutan los “Test cases” que se encuentren vinculados
3. Se visualiza la ejecución de estos en pantalla (se ven las acciones de apertura del enlace y acciones creadas en cada “Test case”)
4. Al finalizar, queda el log de ejecución de los “Test cases” donde se pueden ver los detalles de ejecución y/o los errores en caso de existir

4.4.3 Desarrollo realizado para CEP (*Client Experience Portal*)

Luego de las diferentes planeaciones, asignaciones y tareas realizadas en cada “sprint”, se presenta la automatización final del proyecto, donde tenemos los siguientes datos finales.

Cabe resaltar que para esta aplicación se decidió realizar 2 suites de casos de pruebas:

- Suite de regresión: Sirve para garantizar que las funcionalidades que no se han modificado sigan funcionando conforme a su definición, esta suite se ejecuta

cuando se realiza un despliegue a un ambiente de pruebas (“QA environment”, ambientes bajos)

- Smoke test Suite: Sirve para garantizar que los elementos clave de la aplicación estén funcionando correctamente luego de que se realiza un cambio en la aplicación, esta suite se ejecuta al realizar un despliegue en el ambiente de desarrollo y pruebas (“DEV and QA environment”)

Tabla 2:

Detalle de Automatización Smoke Suite

ACTIVIDAD	Número
Casos de prueba Manuales	49
Casos de prueba Automatizables	19
Casos de prueba No Automatizables	30
Casos de prueba Automatizados	19

En esta tabla se muestra el detalle de los casos de prueba automatizados para la Smoke suite, donde se presentan los casos de prueba automatizados versus los casos de prueba por automatizar y no automatizables.

Tabla 3:

Detalle de Automatización Regresion Suite

ACTIVIDAD	Número
Casos de prueba Manuales	124
Casos de prueba Automatizables	74
Casos de prueba No Automatizables	50
Casos de prueba Automatizados	74

En esta tabla se muestra el detalle de los casos de prueba automatizados para la suite de regresión, donde se presentan los casos de prueba automatizados versus los casos de prueba por automatizar y no automatizables.

4.4.4 Desarrollo realizado para el proyecto DNAV

Para el proyecto DNAV se desarrollaron diferentes suites de ejecución de acuerdo con cada sprint y funcionalidades, al ser este equipo más grande y ser conformado por mínimo 4 integrantes solamente en la parte de automatización de pruebas era necesario tener mayor claridad de que se iba a realizar, para este proyecto se trabajaron en las siguientes tareas:

- Creación de scripts de automatización de acuerdo con la planeación de cada sprint y las funcionalidades a desarrollar
- Asesoramiento y guía a los demás integrantes del equipo
- Organización y auto gestión de tareas en cada sprint
- Creación de “Test Suites” y configuración para su ejecución dentro del pipeline
- Manejo y gestión de los Servidores de ejecución usados en por el pipeline
- Realización de demostraciones en cada sprint
- Soporte y asesoramiento en conocimientos del negocio
- Mantenimientos y mejoramiento de test suites anteriores

Tabla 4:*Detalle de Suites para DNAV*

Suite	Descripción	Número
Smoke	Smoke Test	36
Smoke	Partial Smoke	27
Regression	StandAlone Regression	124
Regression	Exception Analysis	23
Regression	Hand Price - Price Challenge	12
Regression	APP Role – Engagement Role	159
Regression	Procedure Verification	96
Regression	Modularization – Hand Price – Price Challenge - Cancel	11
Regression	OTC validations	54

En la tabla anterior se presenta el detalle del número de test cases automatizados por suites, donde se agrupan por Smoke suite y Regression suites y se describe el módulo o funcionalidad contenido en dicha suite.

4.5 Limitaciones confrontadas en la práctica

Dentro de este proceso de Pasantía se presentaron varias limitaciones, una de las principales fue la comunicación con el equipo de trabajo (equipo interno de la empresa) y con el cliente (equipo de trabajo del cliente), estas limitaciones se presentaron principalmente debido al idioma, ya que la comunicación es en su mayoría se realiza en Inglés. En varias ocasiones se presentaban inconformidades en la comunicación ya que desde uno u otro lado el mensaje que se quería transmitir o recibir no era correctamente entendido. Este es un punto débil que se estuvo trabajando de

diferentes maneras, y que con el paso del tiempo se pudo ir mejorando dicha comunicación, tanto con el afianzamiento del idioma inglés, como con la mejora en la relación de confianza cliente-equipo.

Adicional a esta limitación de la comunicación debido al idioma, también se presentaron inconvenientes en la comunicación interna del equipo (en español) ya que muchas veces el mensaje transmitido no era recibido de la mejor manera. En uno de los equipos de trabajo el rol del pasante era liderar y hacer cumplir ciertas actividades, desde esta perspectiva y la de varios compañeros, se trató de expresar los mensajes de la mejor manera, aun así, uno de los compañeros de trabajo no captaba bien sus labores a realizar. Este punto se estuvo trabajando con ayuda de uno de los líderes de comunicación de la empresa y con el compañero que tenía dificultades de comunicación, esto ayudó a mejorar dicha comunicación y por tanto desde el lado de la empresa se pueden mejorar la comunicación tanto interna, como con el cliente.

Otra de las limitaciones o retos presentados, fue el manejo de personas. Este es un punto muy importante y valioso para la empresa y a nivel profesional, ya que los “soft-skills” (habilidades blandas) deben ser muy bien desarrolladas y aplicadas para tener una buena relación con el equipo de trabajo y con los clientes. Se desarrollaron habilidades como:

Liderazgo, donde por medio de la guía, comunicación y colaboración el pasante logro ayudar a varios compañeros de la empresa a mejorar sus habilidades tanto técnicas como blandas.

Empatía, como se sabe todas las personas tienen diferentes formas de ser, de sentir, de pensar y de actuar. Así que la empatía es uno de los factores más importantes a desarrollar cuando se trabaja con un grupo de personas tan diversos, ya que el equipo de trabajo está conformado por personas de diferentes partes del mundo (India, Ucrania, Estados Unidos, Colombia, y México) fue muy importante para el pasante desarrollar la empatía para mejorar la comunicación.

Autogestión, en los grupos de trabajo que implementan metodologías ágiles, la autogestión es una de las habilidades que todos los integrantes del equipo deben desarrollar, ya que por medio de esta se puede auto organizar y conformar un equipo integral.

Fluidez y confianza en la comunicación en Inglés, al tener un equipo tan diverso el idioma de comunicación es Inglés (idioma universal) que al desarrollar conversaciones y comunicaciones en inglés día a día el pasante pudo adquirir más fluidez y confianza con dicho idioma.

Comunicación, a pesar de las diferentes dificultades que se presentaron con la comunicación en equipo, también se desarrolló y se pudo mejorar dicha habilidad, al solucionar y aprender a manejar las diferencias en la comunicación.

4.6 Aportes del pasante a la Empresa

Con base en la experiencia como pasante dentro de la empresa esta ha sido una de las mejores oportunidades presentadas en la carrera profesional ya que han ayudado mucho a cumplir los objetivos profesionales del pasante debido a que el grado de exigencia está acorde con su personalidad y han ayudado al pasante a exigirse,

ofrecer cada día más de sí y obtener mejores conocimientos tanto personales como profesionales e incrementar mucho su experiencia laboral.

A la empresa se le ha aportado:

- Desarrollo de Pruebas automatizadas para el cliente Deloitte
- Mejoramiento en la calidad del software para el cliente Deloitte
- Soporte al equipo de pruebas con sus conocimientos y experiencias previas
- Formación en la herramienta Tosca Commander a otros compañeros
- Afianzamiento de la confianza Cliente-Empresa
- Mejoramiento de habilidades de comunicación a otros compañeros de trabajo
- Soporte y apoyo a otros compañeros de trabajo
- Liderazgo y ejecución de pruebas automatizadas
- Sus conocimientos, experiencias y estrategias para la ejecución de labores.
- Soporte en la automatización de Pruebas del proyecto Client Experience Portal
- Automatización de Pruebas del proyecto DNAV

4.7 Productos realizados

Dentro de la pasantía se entregaron los siguientes productos:

- Soporte en creación de “Regression suite” automatizada en Tosca Commander para el proyecto “Client Experience Portal”.
- Soporte en creación de “Smoke suite” automatizada en Tosca Commander para el proyecto “Client Experience Portal”.

- Demo realizado en cada “sprint” con la explicación y muestra de los “Test cases” automatizados para el proyecto “Client Experience Portal”.
- Reporte de ejecución del “Regression Suite” por cada “sprint”.
- Reporte de ejecución del “Smoke Suite” por cada “sprint”.
- Diferentes suites automatizadas en Tosca Commander para el proyecto DNAV.
- Apoyo y soporte al equipo de automatización del proyecto DNAV.
- Demo realizado en por cada Suite y creación de “Test cases” en los diferentes “sprints” con la explicación y muestra de los “Test cases” automatizados DNAV.
- Reportes de ejecución de los diferentes “test suites” creados para el proyecto DNAV.
- Soporte técnico en la creación de “Test cases” para el equipo de automatización de DNAV.

Anexos

5.1 Anexo I: Reporte de Ejecución para DNAV

En este [reporte](#) se muestran la ejecución de la “Suite de Regression Standalone” creada para el proyecto DNAV, en este reporte se pretende mostrar tanto una ejecución exitosa, como una ejecución con errores, este reporte es generado con la herramienta Tosca Commander.

5.2 Anexo II: Video de Ejecución en Tosca Commander para DNAV

En este pequeño [video](#) se puede ver el paso a paso de una ejecución en Tosca, donde se muestra tanto la aplicación en Tosca Commander, como la aplicación DNAV que es la que se prueba

5.3 Anexo III: Presentación para Demostración en DNAV

En cada uno de los Sprints se realiza una pequeña demostración, la cual puede ser guiada a través de una [presentación](#) o de un [diagrama](#) y/o ambos, en donde se muestra y se explica que funcionalidad está cubierta en la automatización para dicho sprint y que flujos se tienen contemplados en dicha automatización .

Por motivos de confidencialidad se excluyen algunos elementos de la presentación.

Conclusiones y recomendaciones

Gracias a esta pasantía se pudieron poner en práctica los conocimientos adquiridos en el proceso de aprendizaje de la universidad y también se adquirieron muchos otros conocimientos gracias a los compañeros y a los diferentes retos que se encontraron en el camino de la pasantía.

A la UNAD:

- Mejorar el acompañamiento de los tutores a los alumnos, yo particularmente me sentí muy desescolarizada y de alguna manera abandonada por la universidad, ya que en muchas ocasiones no encontraba solución a diferentes inconvenientes se me fueron presentando en el camino
- Fomentar la realización de Pasantía como opción de grado, ya que esta nos permite a los estudiantes adentrarnos en el entorno laboral y afinar nuestros conocimientos, adicional de adquirir experiencia

A la Empresa:

- Seguir con la formación y afianzamiento de los conocimientos en la herramienta Tosca Commander, ya que es una de las herramientas más usadas por uno de los principales clientes de la empresa y que posiblemente habrá más oportunidades de proyectos con dicha herramienta.
- Seguir con el crecimiento exponencial que ha venido teniendo la empresa

Bibliografía

Amaro Calderón, S. D., & Valverde Rebaza, J. C. (2007). Metodologías ágiles. *Perú: Trujillo*. https://www.academia.edu/download/53222887/Metodologias_Agiles.pdf

Azure.microsoft.com, (2020). Azure DevOps [En Línea]
<https://azure.microsoft.com/es-es/services/devops/#overview>

Calvo-Manzano, J. A., Garzás, J., Piattini, M., Pino, F. J., Salillas, J., & Sánchez, J. L. (2008). Perfiles del ciclo de vida del software para pequeñas empresas: los informes técnicos ISO/IEC 29110. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 4(2), 96-108.

<https://www.redalyc.org/pdf/922/92218339010.pdf>

Hepner, L. (2019). El desafío de la agilidad en las organizaciones: ¿ un nuevo mindset para la transformación de los negocios?.

<http://dspace.aepro.com/xmlui/handle/123456789/2428>

Martínez, J. I., & Comino López, M. (2018). El método ágil Scrum, evolución y aplicación en la administración de proyectos.

<http://dspace.aepro.com/xmlui/handle/123456789/1562>

Mascheroni, M. A., & Irrazábal, E. (2018). Problemas que afectan a la Calidad de Software en Entrega Continua y Pruebas Continuas. In *XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018)*.

http://sedici.unlp.edu.ar/bitstream/handle/10915/73270/Documento_completo.pdf-PDFA.pdf?sequence=1

Radigan, Dan, (s.f.). Reuniones rápidas para equipo ágiles

<https://www.atlassian.com/es/agile/scrum/standups>

Trigas Gallego, Manuel. Metodología Scrum. [En Línea].

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

Van Lancker, L. (2013). *Los API JavaScript de HTML5: Integre la potencia de HTML5 en sus aplicaciones Web*. Ediciones ENI.

<https://www.academia.edu/download/35567868/APIJAVA.pdf>