

Universidad de Lima
Facultad de Ingeniería
Carrera de Ingeniería de Sistemas



DESARROLLO PROFESIONAL HACIA EL “FUNCIONAL TESTING” Y ANÁLISIS DEL NEGOCIO

Trabajo de suficiencia profesional para optar el Título Profesional de Ingeniero de
Sistemas

Jesus David Quispe Casas

Código 19942519

Asesor

Jorge Luis Irey Nuñez

Lima – Perú

Mayo de 2023



**PROFESSIONAL DEVELOPING FORWARD
TO DOMAIN OF TESTING IN BUSINESS
ANALYSIS**

TABLA DE CONTENIDO

RESUMEN	vi
ABSTRACT.....	vii
INTRODUCCIÓN	1
1. CAPACIDAD TÉCNICA	3
2. CAPACIDAD DE GESTIÓN	17
3. APRENDIZAJE CONTINUO.....	30
3.1 Tópicos de aprendizaje	31
4. CONDUCTA ÉTICA	45
5. LECCIONES APRENDIDAS.....	47
6. GLOSARIO DE TÉRMINOS	50
REFERENCIAS.....	54
BIBLIOGRAFÍA	56

ÍNDICE DE TABLAS

Tabla 1.1 Fases del proyecto de ampliación de espacio en servidores	4
Tabla 2.1 Justificación para la mejora de la plantilla y generador de reportes de ejecución de pruebas.	21



ÍNDICE DE FIGURAS

Figura 1.1 Línea de tiempo de mi experiencia laboral	3
Figura 1.2 Proceso de control de calidad de software	6
Figura 1.3 Ciclos de despliegues a producción del portal Web, backend APP	10
Figura 1.4 Pasos para automatizar	12
Figura 1.5 Entorno de Automatización.....	13
Figura 1.6 Arquitectura del APP.....	14
Figura 1.7 Repositorio de casos de prueba en JIRA	15
Figura 1.8 Repositorio del Postman.....	15
Figura 2.1 Cálculo del número de casos de prueba entregables para Biblioteca.....	17
Figura 2.2 Plantilla de Biblioteca	18
Figura 2.3 Plantilla Resumen de Biblioteca.....	18
Figura 2.4 Flujos y sus casos asociados.....	19
Figura 2.5 Reportes Gerenciales	20
Figura 2.6 Presentación de la Plantilla.....	22
Figura 2.7 Codificación de casos de prueba	23
Figura 2.8 Reporte de ejecución	23
Figura 2.9 Plantilla de Evidencias generadas para un proyecto en específico	24
Figura 2.10 Reporte Exportado del Mantis BT.....	24
Figura 2.11 Plantilla para la bitácora de incidencias	26
Figura 2.12 Inventario de Microservicios utilizados	28
Figura 2.13 Inventario de casos de prueba por microservicio	29
Figura 3.1 Business Model Canvas.....	40

RESUMEN

Ser egresado de la Universidad de Lima me ha permitido adquirir una serie de experiencias tanto en lo laboral como en lo personal a las que yo consideraría como invaluable. Por cada rol o puesto de trabajo que tuve que realizar, iba acumulando una serie de habilidades, conocimientos y aptitudes, necesarias para desempeñar tal rol y otras funciones que se iban agregando conforme a la confianza ganada. Los conocimientos adquiridos al convertir los problemas en oportunidades han sido lo que más satisfacción me han proporcionado, ya que sus frutos son herramientas tangibles para el desarrollo de nuevas actividades.

A lo largo de estos años mi desarrollo en las áreas y posiciones, como analista programador, consultor de sistemas, funcional, control de calidad de software, ingeniero de software nivel 1 y, líder de un equipo de “testing”, también me permitieron adquirir habilidades no solo técnicas sino también las denominadas “habilidades blandas”, muy importantes para que junto con las habilidades técnicas pueda guiarse a través de la comunicación, empatía, confianza y otros, así lograr mejoras profesionales como ascensos, bonos por productividad e incluso reconocimientos por desempeños y logros de objetivos.

Luego de haber logrado un esperado ascenso y ser líder de un equipo de “testing”, el logro de objetivos, cumplimiento de las tareas, liderar y gestionar recursos humanos no era una tarea sencilla, no todas las personas tienen la misma motivación y compromiso en el cumplimiento de sus tareas diarias, y es en estas situaciones que debemos que aplicar nuestras habilidades blandas para lograr una mejor comunicación, poder negociar y ser la guía para que nuestros colaboradores puedan lograr también sus objetivos en beneficio del proyecto

Para terminar, ser egresado de la facultad de Ingeniería de Sistemas de la Universidad de Lima nos permite crecer y adquirir lo necesario para que junto con ciertas aptitudes se pueda alcanzar el éxito que deseamos.

Palabras clave: Error, defecto, fallo, API, microservicio, emulador, ISTQB, simulador.

ABSTRACT

Being graduated from Lima University has allowed me a group of experiences as well at work and personal, that I will consider as “priceless”. For every role and job position that I had to do, I was gathering a set of abilities and knowledgement, necessary for such a role and other functions according to the confidence earned.

Along these years, my develop in certain areas and positions, as analyst developer, system consulting, functional and software quality control, software engineer level 1 and finally Testing Team Leader, also allowed me acquire not only technical skills but also “soft skills”, very important because along with the technical ones may guide through communication, empathy, confidence and others to get professional improvements, promotions, bonus and recognitions for performance, productivity and goals achievements.

After achieving the expected promotion for being the Testing Team Leader, goals achievement and tasks accomplishment and leading and managing human resources was not an easy task, not all the people have the same motivation, commitment in accomplishing daily tasks. It was right there where I had to use all the soft skills for communicating, negotiating and to be the guide for our collaborators may achieve their goals in benefit of the project and company.

Finally, graduating from the System Engineering Faculty from University of Lima provides you the tools and opportunities that allow you to grow and acquire the necessary skills to achieve the success that we wish.

Keywords: Error, bug, fail, API endpoint, framework, automation, ISTQB, emulator.

INTRODUCCIÓN

Haciendo una retrospectiva de mi vida laboral y por todo lo vivido en la diversidad de áreas que me he desenvuelto por cada experiencia que me ha aportado en lo profesional y personal, se podría resumir o explicar cómo lo menciona Lucas Estrella Shultz en varios pasajes de su libro “El Oráculo del guerrero”, encuentro palabras que serían una analogía a mi vida profesional como “un camino en constante evolución”...”A lo largo de su vida un guerrero pasa por diferentes situaciones, etapas y pruebas” y para terminar con lo que se menciona al inicio de su capítulo III “El Guerrero emprende el viaje” donde el autor menciona y se podría tomar como recomendación :

Debemos marcharnos cuando el entorno ya no aporta nada nuevo, cuando nos encontramos estancados, que dejemos lo superfluo, a esto puedo describir como ciertas comodidades que el empleador te ha otorgado para poder retenerte, pero sin expectativas de crecimiento profesional.

El autor finaliza con el párrafo “Tu corazón, tu espíritu y tu cuerpo son suficientes”, y esto es lo que se debería pensar por cada partida de cada empresa con la misma fe de siempre, que mi forma de trabajar, el esfuerzo y los deseos de aprender que siempre pongo ayudan a desarrollarme en el nuevo puesto laboral, también es importante lo que menciona en el siguiente párrafo del siguiente capítulo, y es que debemos dejar todo en orden, no romper puentes para que podamos volver, ya que todo cambia con el tiempo.

La capacidad analítica, del Ingeniero de sistemas, permite entender el funcionamiento de cada componente de un sistema y a su vez entender las consecuencias de algún malfuncionamiento de sus componentes, es aquí donde el Control de Calidad de Software interviene, si definimos al sistema como un aplicativo o software desarrollado a medida y con gran impacto en el negocio o en la vida de las personas; entonces como profesional de esta área al analizar los posibles fallos de cada componente se pueden prevenir mal funcionamientos y evitar fallas que algún caso pueden ser hasta catastróficas y que podrían como consecuencia, pérdidas millonarias para el negocio.

Desde mi graduación de la carrera de Ingeniería de sistemas incluso desde mi periodo como practicante, me he desenvuelto como analista programador, para luego como egresado continuar como analista o consultor junior de TI, administrador de pequeños negocios, analista funcional ante la necesidad del proyecto por cierta coyuntura en la relación Cliente Proveedor. Son los roles de Consultor Junior de TI y analista en el diseño y desarrollo de sistemas que son el soporte para mi rol de mayor duración como Analista de control de calidad, el cual ya vengo desarrollando hasta la fecha, entregando soluciones libres de defectos o llevando riesgos al mínimo. Adicionalmente a las habilidades técnicas adquiridas, también se me ha permitido desarrollar y adquirir habilidades blandas mediante talleres proporcionados por mis empleadores como parte del desarrollo y capacitación de sus empleados; es así como en el campo de Control de Calidad, lo que más se necesitan son la empatía, confianza, comunicación y colaboración muy importante para liderar equipos dentro del área y tratar con diferentes tipos de actitudes.

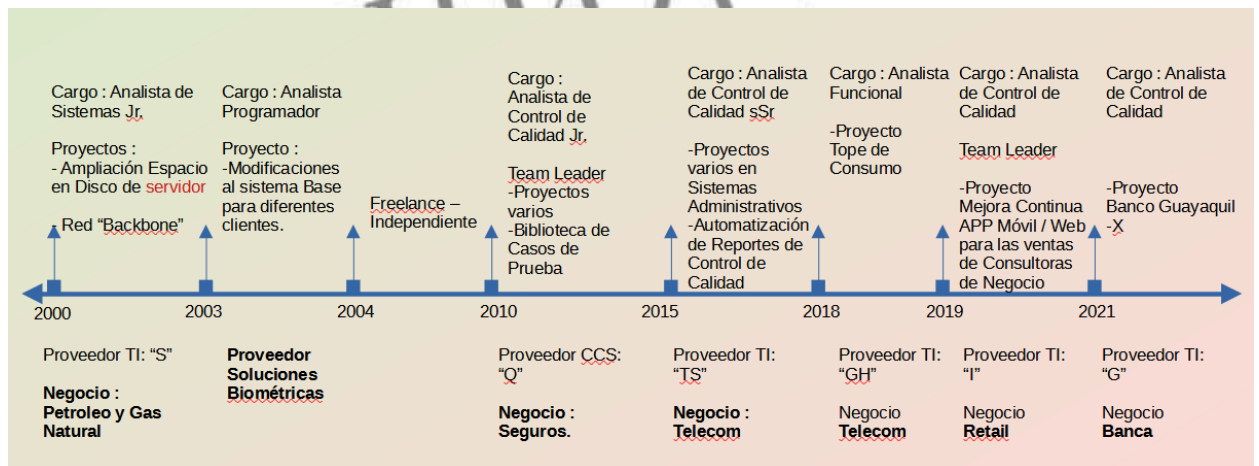
Es principalmente mi trabajo como Analista de Control de Calidad de Software lo que más quiero detallar en este informe, porque es el reflejo de lo vivido en experiencias anteriores y que contempla una serie de habilidades tanto técnicas como blandas.

1. CAPACIDAD TÉCNICA

Desde que egresé de la carrera de Ingeniería de sistemas he adquirido experiencia y conocimientos en varias áreas y tipos de negocio como se muestra en la **Figura 1.1**

Figura 1.1

Línea de tiempo de mi experiencia laboral



En los comienzos de mi experiencia laboral, como analista junior de TI en el 2000 participé en un proyecto para la ampliación de capacidad de almacenamiento en los servidores como parte de implementación de mejoras en la red de la compañía, ante la problemática que las fallas en las estaciones de trabajo de cada usuario (el más común era problema de disco), ocurrían con más frecuencia y la información de los usuarios, información sensible de la empresa poseía poco o no se tenía respaldo.

Para llevar a cabo la implementación se realizó según lo que se muestra en la **Tabla 1.1**.

Tabla 1.1

Fases del proyecto de ampliación de espacio en servidores

Estudio y preparación	<p>En esta etapa se planificó de se iba a tomar los datos de cada usuario por piso / área.</p> <p>Se elaboraron fichas para ser llenadas en el sitio de cada usuario por el analista asignado</p>
Investigación – Encuesta	<p>Se lleva a cabo la toma de datos de cada usuario y la cantidad de espacio usado por los documentos trabajados con información de la empresa en sus PC asignadas.</p>
Recopilación de datos	<p>Obtenido los datos de todos los usuarios, se clasifican por el área y por piso y se tabularon los datos en tablas.</p>
Reporte y análisis	<p>Se obtiene el reporte a partir de los datos obtenidos sobre el almacenamiento ocupado por cada usuario en sus respectivos equipos.</p>
Toma de decisiones para la infraestructura y tecnología a implementar.	<p>Desde la jefatura de Sistemas se tomaron las decisiones para la adquisición de la solución a implementar a partir del resultado de los reportes.</p> <p>A partir de los análisis de los reportes se hace la investigación de tecnología y proveedores que brinden soluciones de servidores y plan de soporte con las garantías requeridas.</p>
Implementación	<p>Se implementó el sistema de almacenamiento para la red de la compañía.</p> <p>Se crearon las carpetas para que cada usuario con sus credenciales tuviese acceso a las carpetas creadas.</p> <p>Se crearon carpetas compartidas para que los usuarios de ciertas áreas que usaban el mismo archivo pudiesen realizar las modificaciones requeridas.</p>
Migración de la Data	<p>Ejecutado la implementación e instalación de la infraestructura solicitada se comunicó a los usuarios que procedan a migrar la data hacia sus carpetas asignadas en el servidor.</p>
Monitoreo	<p>En esta etapa se monitorea el rendimiento del servidor en los picos de migración de la data de las PC al servidor.</p>

Como sistema de almacenamiento seleccionado por la jefatura de sistemas se gestionó con los proveedores seleccionados para la implementación el Sistema de discos RAID 5 Arreglo redundante de discos independientes o Redundant Array of Independent Disk (RAID), estos sistemas de almacenamiento se caracterizan por su gran tolerancia a fallos ya que cuentan con su propia contingencia de usar un disco como “backup” (copia

de respaldo) o “mirror” (espejo) de los otros, y al malograrse uno, al final solo se necesita reemplazarse.

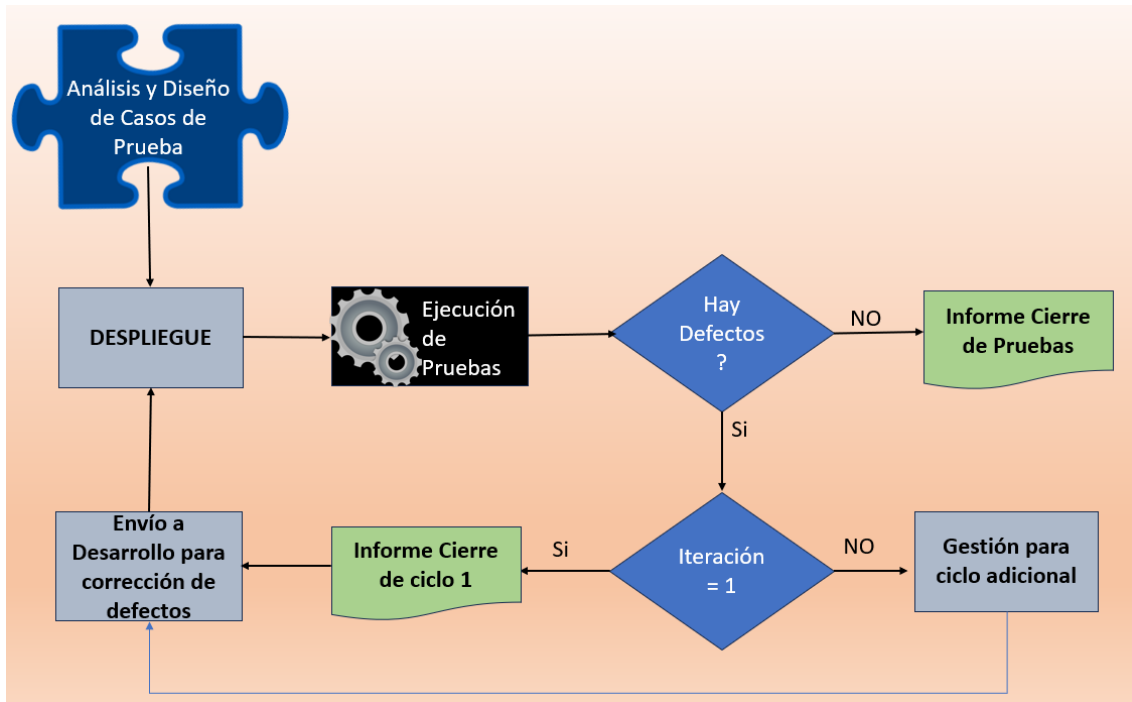
Otro de los proyectos como parte de las mejoras planificadas en la infraestructura tecnológica por expansión del site (se reubicaría a los miembros del proyecto de gas natural ya que en ese momento se encontraban en otro local) se implementó el proyecto de un nuevo Sistema de red de datos “backbone” para distribuir redes virtuales a cada piso (3 pisos inicialmente, luego se agregó un piso más), esta implementación previa facilitaba la adición de áreas completas o estaciones de trabajo individuales tanto para personal externo como personal de la compañía. Firewalls y antivirus también fueron considerados en las mejoras la protección de las nuevas implementaciones realizadas.

Antes de iniciar mi vida laboral en el Control de Calidad, tuve un paso como independiente desarrollando diversos trabajos, el primero como Analista programador para una empresa de soluciones biométricas donde el conocimiento y experiencia adquirida en lenguaje VBA, fue útil muchos años después. Posteriormente a manera de freelance, fui dando consultoría, administración y soporte a los recursos tecnológicos usados en una clínica pediátrica.

A mi regreso de una experiencia en el extranjero, en el 2010 me inicié en el campo de Control de Calidad de Software perteneciendo a una empresa dedicada al rubro, como analista junior de Control de Calidad de Software, asignado al cliente, una reconocida compañía de seguros, donde desarrollé mis principales actividades que se pueden apreciar en la **Figura 1.2**.

Figura 1.2

Proceso de control de calidad de software



Todas estas actividades se realizaron para los diversos proyectos que se requerían poner en producción por mejoras a realizar, digitalización de procesos, regulaciones de la SBS.

- Flujo de emisión de pólizas de Vida y Accidentes personales
 - Preventa. (Cotizador de seguros de Vida y accidentes personales)
 - Digitación y registro
 - Evaluación
 - Cobranzas
 - Emisión, impresión y entrega de pólizas.
- Pagos a los Asesores de Seguros: Compensaciones, Cálculo de comisiones por ventas.
- Administración de Pólizas de Vida Grupo en sus diferentes líneas
 - Vida Ley, Vida Ley Obreros, SCTR, Vida Grupo.

Para todos los sistemas desarrollados tanto para Web (aplicativos. net), como aplicativos Windows (Visual Basic 6, .net) era necesario verificar el cumplimiento de las funcionalidades desarrolladas de acuerdo lo requerido, algunas de estas funcionalidades venían también acompañadas de aplicación de reglas de negocio, necesarias tanto para la toma de decisiones de los usuarios de estas aplicaciones como para el cumplimiento de regulaciones y/o información que se le proporcionaba al cliente desde la preventa hasta la entrega de su póliza.

La no verificación de algunas funcionalidades venía con el riesgo de algún incumplimiento en reglas del negocio, y/o asuntos gubernamentales, y que conlleva a demandas millonarias y sanciones por parte de entidades gubernamentales

Adicionalmente a los proyectos donde teníamos participación se llevó a cabo una Biblioteca de Casos de Prueba, el cual consistía en la recopilación de los casos o creación de casos de determinadas aplicaciones que se iban usando en los procesos de pruebas de determinados proyectos.

Esta biblioteca era desarrollada cuando:

- El analista no tenía una asignación
- A la espera del despliegue de la solución para el inicio de la ejecución de pruebas
- A la espera del despliegue de la solución con los defectos solucionados

Mientras sucedía lo mencionado se debían desarrollar actividades de Biblioteca y así justificar las horas no consumidas del proyecto dentro del horario laboral, la forma de gestionar la biblioteca la detallaré en la sección de Capacidad de Gestión.

Tras 5 años de aprendizaje en el campo de los seguros, cambié de empresa proveedora de soluciones tecnológicas, cuyo cliente era una gran compañía de telecomunicaciones dónde el equipo de calidad era parte del equipo de desarrollo.

Dentro de los proyectos más importantes dónde se realizaba las actividades de control de calidad estaban involucrados los siguientes tipos de aplicativos Web desarrollados en .Net

- Sistemas Administrativos

- Sistemas de Ventas
- Sistemas de Post venta.

Cabe resaltar que dentro de los sistemas administrativos se desarrolló un proyecto de refactorización de código conforme a los estándares de OWASP.

Cuando inicié mis actividades en el equipo de control de calidad, en el ejercicio diario con el transcurrir de los días se observó que debíamos realizar un corte con mínimo 2 horas antes del cierre del día (5:00 de Lunes a Jueves y 4:30 los viernes) para emitir los reportes diarios y reporte semanal, este corte con exagerada antelación era debido a cantidad de información recopilada de forma manual, durante la ejecución tanto como casos ejecutados como las “incidencias” encontradas durante la ejecución. Si bien el registro de casos exitosos era manual, la asociación de las incidencias encontradas y registrados en el Mantis BT con los casos ejecutados y la posterior elaboración del reporte de ejecución era una tarea diaria y tediosa para poder obtener los reportes.

Las herramientas para llevar la ejecución diaria era una plantilla previamente confeccionada y que era de uso para cada proyecto y por medio de una “macro” en excel, los casos solo eran codificados y los “bugs” se registraban en una herramienta “Bug tracker” llamado Mantis BT, asociándose al caso ejecutado, esta herramienta era usada como herramienta de comunicación entre el equipo de calidad y el equipo de desarrollo, dónde se informaban los bugs nuevos, así como los estados transitorios hasta su cierre.

Las fases del proceso de prueba llevadas a cabo para cada proyecto eran las siguientes:

- Análisis: a partir de la documentación generada por el analista funcional y desarrolladores del proyecto.
- Diseño de los casos de prueba: Se desarrollaban los casos de prueba empezando por la pestaña de Pruebas Funcionales la cual contemplaría casos Ok y noOk, es decir pruebas a detalle con la verificación de la UI. Una vez terminado el diseño de los casos de prueba en la pestaña de pruebas funcionales se procede a analizar qué casos deberíamos contemplar para “smoke test” y evaluar la estabilidad inicial del producto desarrollado, finalmente el diseño de las pruebas de regresión se daría terminada las pruebas funcionales mediante

una revisión seleccionando los casos de aquellas funcionalidades que fueron impactadas durante el hallazgo y resolución de defectos (“bugs”) en la ejecución de pruebas funcionales.

- Como parte de esta etapa también se preparaban los reportes con las funcionalidades, subfuncionalidades y casos diseñados tanto para el avance de la ejecución como para la bitácora de incidencias.
- Ejecución: Una vez terminado el producto o ciertos módulos del producto se nos entregaba a los analistas de calidad para que iniciemos con el proceso de ejecución de pruebas, es decir verificar que el producto haga todo lo que se indicó en el documento de análisis y especificaciones, de no cumplir cierto requerimiento se registraban la(s) incidencia(s) en el Mantis BT, relacionando la incidencia con el caso de prueba.
- Al terminar las operaciones del día se hacía el recuento de las incidencias y se empezaban a relacionar con los casos en la plantilla, para obtener un reporte.
- Cierre de las pruebas: con la ejecución terminada y el producto estabilizado se procede con la elaboración del informe de cierre, para que el producto luego pueda ser entregado a la entidad certificadora antes de que sea puesto en producción.

En el 2019 cambié de rubro de negocio a “retail” siendo la primera vez en pertenecer a un equipo de mejora continua donde se usaba Scrum como marco de trabajo para el desarrollo ágil y donde el principal activo digital del equipo era el “app” desarrollado en el sistema operativo “Android” soportado también por el aplicativo WEB (donde otro equipo era el responsable) y destinado para el uso de las consultoras de este “retail” para la recepción de pedidos de sus clientes, ventas y facturación.

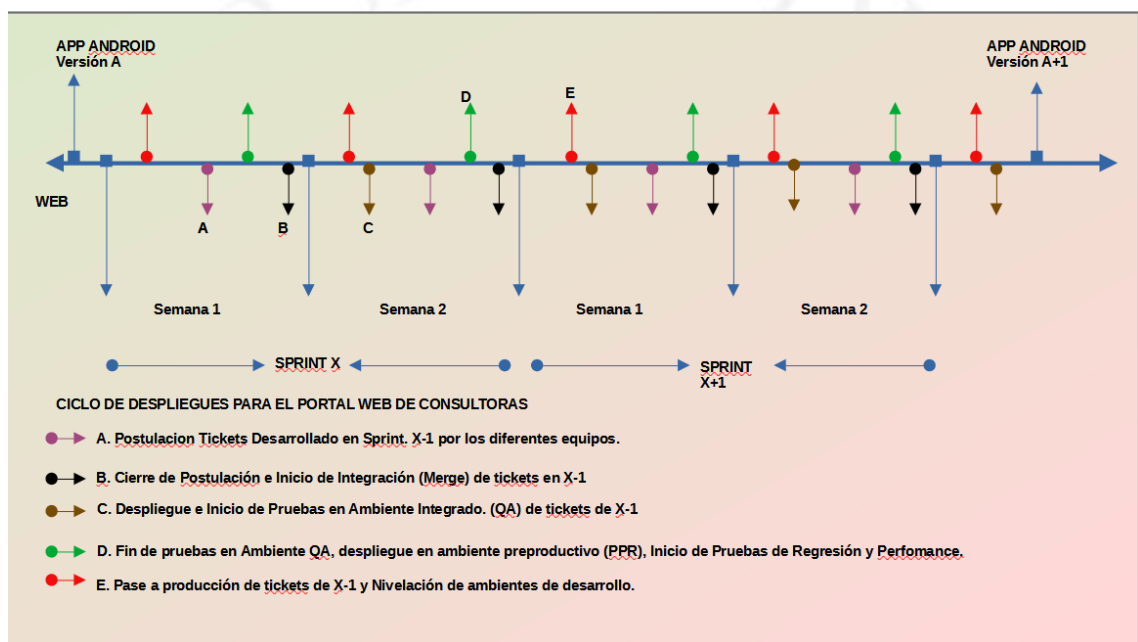
Las consultoras del negocio tenían a su disposición como herramienta de venta tanto el sistema WEB como el APP de Android. Cabe resaltar que para estos activos digitales había más de un equipo que se encargaba de otras funcionalidades involucradas, y cada equipo ingresaba sus códigos a través de los “pull request” donde un líder técnico responsable se encargaba de realizar las integraciones de código y así desplegar una sola

versión unificada para web y una sola versión para la app para las diferentes fases de promoción entre los ambientes de prueba de cada activo.

En la **figura 1.3** se indica como los periodos de entrega a producción de los requerimientos para el sistema WEB (desktop y mobile) son en periodos semanales y para el APP (Android) era de manera mensual, por tanto, el desarrollo de las mejoras para los microservicios involucrados en cada pase a producción del sistema WEB requería de pruebas de regresión en la versión actual de Android.

Figura 1.3

Ciclos de despliegues a producción del portal Web, backend APP



Todos estos despliegues del sistema WEB y el APP podían ser monitorizados a través del sistema de “DEVOPS”.

En el caso post pase a producción, se encuentre alguna incidencia en Producción crítico, el ciclo de despliegue se abrevia, haciéndose un duplicado del ambiente pre productivo nombrado como PPR FIX, se soluciona el problema o el incidente y se realizan las pruebas por el equipo responsable del ticket que ocasionó el incidente y se pasa a producción, una vez obtenido el visto bueno de los usuarios se procede con la nivelación de todos los ambientes.

De las principales funcionalidades del APP y críticas para el negocio tenemos:

- Ingresar Pedidos: Ingresando a esta funcionalidad, se visualiza la pantalla llamada “Pase de pedido”, Las consultoras colocaban sus pedidos, algunos

productos eran agregados automáticamente por promociones u ofertas y al cierre de fecha de campaña el pedido era facturado y enviado a procesamiento para entregar a la consultora los ítems solicitados.

- Catálogo Digital: Las consultoras podían compartir el catálogo digital a sus clientes para que a través de un enlace los pedidos del cliente sean colocados en el Pase del Pedido
- Tienda Online: Herramientas de ventas de la consultora para ofrecer más variedad de productos y ofertas, realizar una venta y entrega directa de la Consultora hacia sus clientes.
- Pago en línea: A través de esta funcionalidad, la consultora realizaba el pago de los productos solicitados en la campaña recién facturada y cerrada.
- Gana+: Funcionalidad donde la consultora tenía más ofertas de productos y podía ofrecer a sus clientes, paquetes o kits de productos, esta funcionalidad era responsabilidad del equipo de trabajo del mismo nombre.

En esta etapa de mi experiencia laboral adquirí experiencia, para realizar actividades de “testing” en equipos móviles, adicionalmente mediante iniciativa y de la comunidad QA “in site” del cliente mediante capacitaciones periódicas, se me permitió las ventajas de la automatización de pruebas en un “framework” amigable llamado “codeceptjs”, sin embargo, la preparación del ambiente para empezar a desarrollar la automatización consistía de instalación de múltiples herramientas, “drivers” y otros.

- Visual Studio Code: IDE o entorno de programación para escribir el código de automatización.
- Entorno: Node Js, Android Studio.
- Extensión: Cucumber, herramienta que soporta la programación a partir del Behavior Driven Development (BDD), donde se usa una estructura de lenguaje llamada Gherkin, el cual permite especificar comportamientos esperados del software que los usuarios finales puedan entender.
- Helpers: Puppeteer, Protractor, WebdriverIO, estos contienen grupos de funciones que hace más sencillo la automatización en vez de estar desarrollando las funciones desde 0. Quiere decir que la automatización de

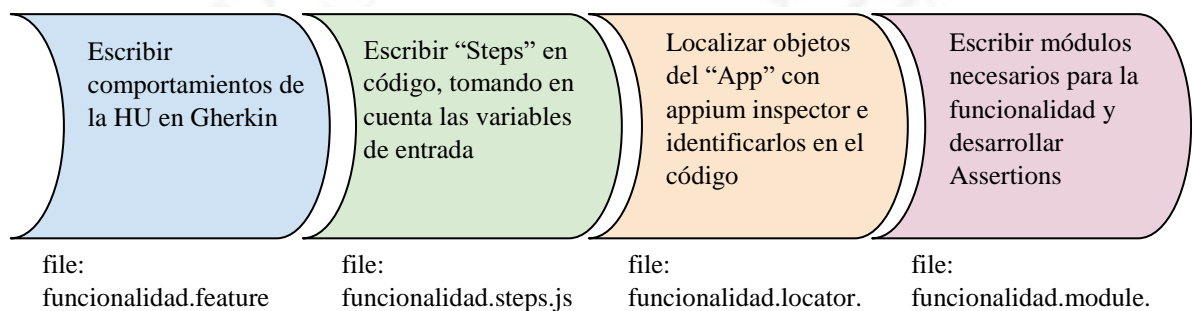
pruebas de una aplicación puede terminarse en menos tiempo con la ayuda de estos “helpers”.

- Appium: Servidor y Localizador de objetos para aplicaciones Android.

Instaladas las herramientas anteriormente descritas y conociendo un poco de lenguaje “Gherkin” seguimos los siguientes pasos como se indica en la **Figura 1.4**

Figura 1.4

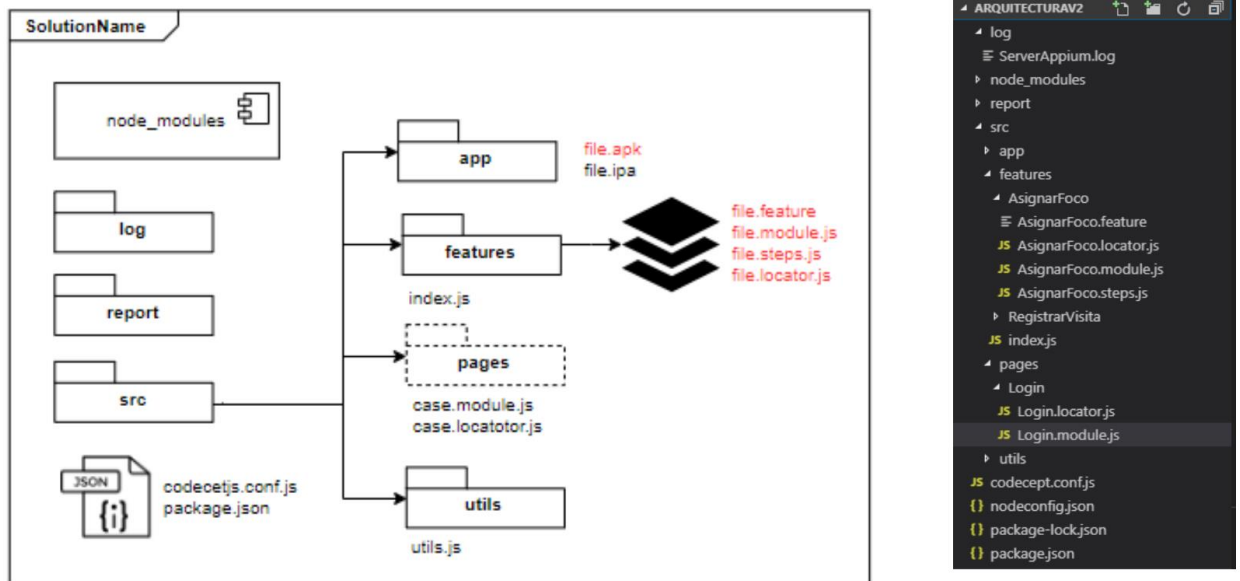
Pasos para automatizar



Bajo el modelo de desarrollo de objetos por página (Page object model [POM]) se escribe código por cada pantalla invocada por cada objeto del APP para realizar una función requerida. Cada pantalla tendrá su grupo de archivos descritos en la **Figura 1.4** y de ocurrir un error durante la ejecución de la automatización será muy sencillo identificar dónde ocurrió el error, un ejemplo del entorno en el que se escribe código para automatizar pruebas es lo que se muestra en la siguiente **Figura 1.5**.

Figura 1.5

Entorno de Automatización



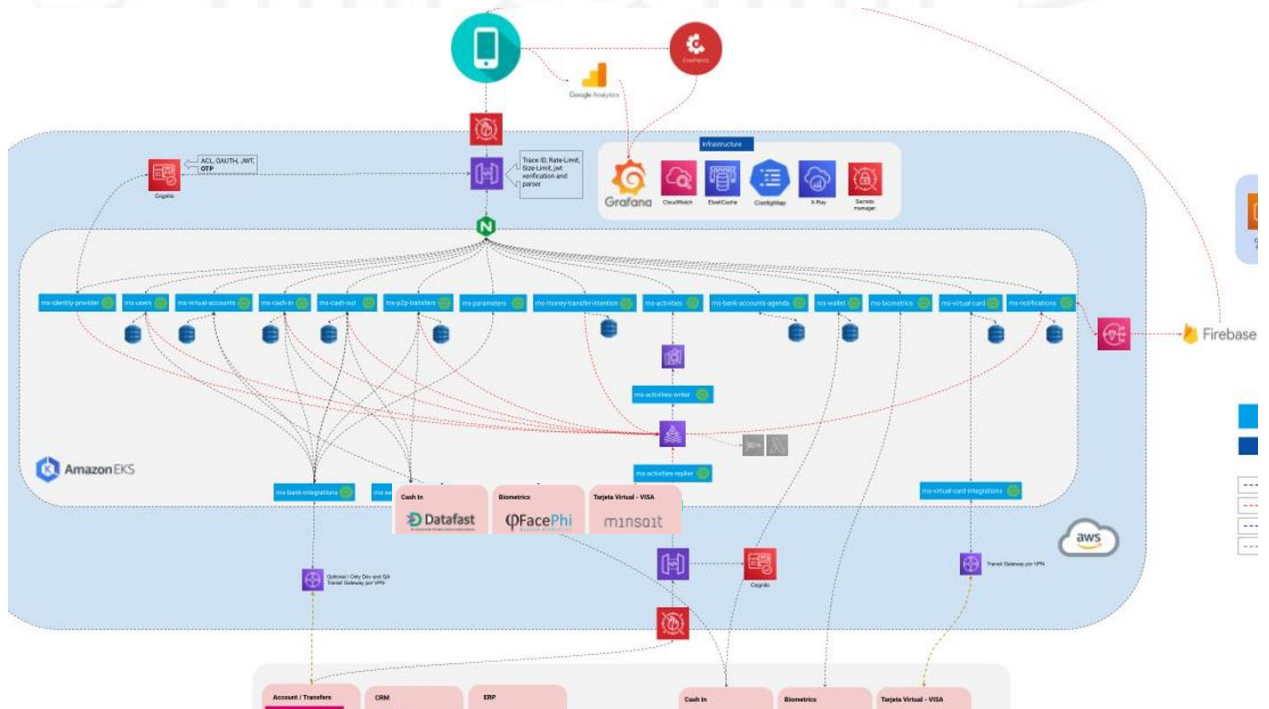
Después todo lo explicado anteriormente, si bien la automatización no era mi función principal en el proyecto y en el equipo de mejora continua, se desarrolló la automatización de ciertas funcionalidades como: “login” con distintos escenarios , colocar productos en un pedido y el guardado la orden, Ingreso a cada opción de la pantalla principal y a cada opción del menú, si bien esta automatización era ejecutada de manera local desde una PC y no desde la nube dentro del ciclo “ DEVOPS” sirvió para verificar funcionalidades de ciertos requerimientos y ejecutar pruebas de regresión de sobre diferentes versiones del sistema operativo Android.

En la medida que mi asignación a este proyecto transcurría en el tiempo, fui conociendo más del negocio y como el APP estaba desarrollada, con sus configuraciones en otras herramientas por ejemplo Base de datos, REDIS, New RELIC, Administrador de Contenidos, su conexión con el sistema comercial y otras herramientas de venta accesibles desde el app, con todo este conocimiento, empiezo a realizar actividades de liderazgo, distribución y asignación del trabajo y formación de nuevos miembros del equipo de analistas de control de calidad en el funcionamiento del App/Web , funcionamiento del negocio, el ciclo de desarrollo y pases a producción tanto de la web como del app. Al observar que ya contaba con un equipo con Analistas ya empoderados en sus tareas con el conocimiento necesario decidí ir en busca de nuevas expectativas laborales.

Es por eso por lo que en el 2021 cambié de empleador, siendo asignado a un proyecto de una billetera electrónica para un banco del extranjero como analista de control de calidad para un proyecto nuevo y sin lineamientos para el inicio con el equipo de calidad nos empezamos a organizar para establecer estos lineamientos, desde los artefactos y sus nomenclaturas, procesos, repositorios.

Con el paso del tiempo, luego de muchos “sprints” desarrollados en el proyecto se contaba con una cierta cantidad de microservicios donde cada analista ya tenía cierto conocimiento, vi la oportunidad de agrupar lo ya desarrollado en un repositorio común dentro de la herramienta postman, con automatizaciones en asignaciones de variables de entorno, colecciones (conjunto de “requests” o llamadas al servicio organizados en carpetas), locales y test automatizados para obtener las respuestas deseadas al ejecutar las operaciones de estos microservicios según la especificados en los documentos llamados “contrato”, todos estos microservicios vienen a ser la base para la distintas operaciones y funcionalidades desarrolladas en el APP

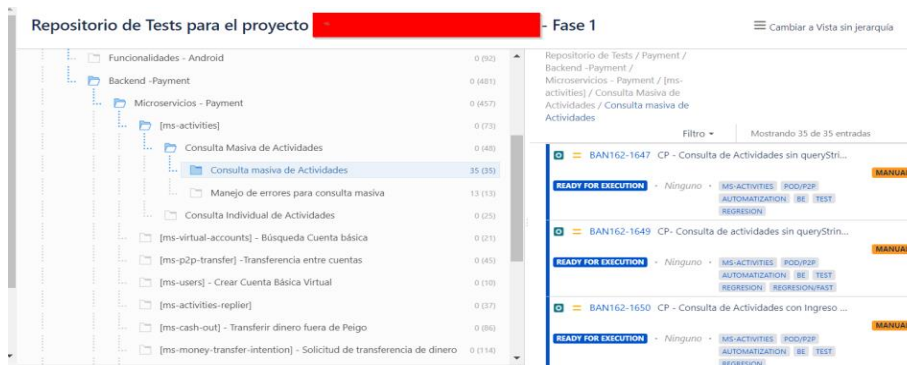
Figura 1.6
Arquitectura del APP



En caso de que en un sprint en determinada historia de usuario se modificaran algunas características de un microservicio, se tendrían que tomar los casos necesarios organizados en el repositorio de casos de prueba dentro del Jira según la funcionalidad o microservicio que se muestran en el panel derecho de la **Figura 1.7**.

Figura 1.7

Repositorio de casos de prueba en JIRA



Realizadas las modificaciones necesarias, en los casos de prueba, también se procede con la actualización del repositorio del postman, agregando o modificando las validaciones necesarias a manera de “automatizar pruebas” en la pestaña “tests” como indica la **Figura 1.8**

Figura 1.8

Repositorio del postman



Junto con el equipo se actualiza el inventario por cada “feature” desarrollado o modificado y con las pruebas ya realizadas de manera satisfactoria, ya que, de encontrarse algún defecto, se tendría que esperar por la corrección para así actualizar los inventarios con desarrollos libres de defectos.

Todo esto permite que, en las pruebas de regresión en entornos preproductivos y dedicados a pruebas de usuario : User Acceptance Test (UAT) se realicen de una manera “ágil” y con alto grado de confiabilidad para la entrega rápida al público.



2. CAPACIDAD DE GESTIÓN

Si bien mi experiencia en la gestión ha sido breve, ser miembro proactivo de los equipos donde he pertenecido me ha llevado a desarrollar mejoras e implementar soluciones en los procesos.

Como analista de control de calidad de software empezaré a detallar mi primera actividad de gestión que consistía en las entregas de la biblioteca de casos de prueba.

Era mi trabajo organizar esta biblioteca y verificar que se entreguen el número de casos obligatorios luego de entregar la contabilización del mes versus horas reportadas al proyecto. La forma de calcular el trabajo de “biblioteca” a entregar se realiza con un factor de 10 casos por hora no asignada a un proyecto, en el ejemplo de la **Figura 2.1** se indica cuántos casos de biblioteca se podría entregar en un mes.

Figura 2.1

Cálculo del número de casos de prueba entregables para Biblioteca

Mes : Junio						
Días laborales del mes	22					
Horas Obligatorias por día	9					
Total de Horas obligatorias al mes	198					
Ratio Casos por Hora	10					
Dcto. por Descansos Médicos	18					
Horas. Asignadas a Proyecto						
	Proyecto 1	Proyecto 2	Proyecto 3	Total Horas Asignadas a Proyectos en el Mes	Horas no Asignadas	Número Aproximado de casos por entregar a Biblioteca
Analista 1	48	65	80	193	5	50
Analista 2	36	50	75	161	37	370
Analista 3	60	35	96	191	7	70
Analista 4	75	53	0	128	70	700
					Dcto de casos	-180
					Total de Casos a entregar para Biblioteca	1010

Como parte del trabajo de biblioteca se utilizaba como artefacto una plantilla de casos de prueba por aplicativo como se muestra en la **Figura 2.2**

Figura 2.2

Plantilla de Biblioteca

TC = 100		NOMBRE MODULO1								Resultado = 0	Avance = 0%			
Título	Nro	Sub Modulo	Opción	Función	Caso de Prueba	Data de Prueba	Usuario / ROL	Resultado Esperado	Fecha	Resultado	Avance (%)	2do Ciclo	Resultado	Proyecto
Casos	1									0	0%	0		
Casos	2									0	0%	0		
Casos	3									0	0%	0		
Casos	4									0	0%	0		
Casos	5									0	0%	0		
Casos	6									0	0%	0		
Casos	7									0	0%	0		
Casos	8									0	0%	0		
Casos	9									0	0%	0		
Casos	10									0	0%	0		
Casos	11									0	0%	0		
Casos	12									0	0%	0		
Casos	13									0	0%	0		
Casos	14									0	0%	0		
Casos	15									0	0%	0		
Casos	16									0	0%	0		
Casos	17									0	0%	0		
Casos	18									0	0%	0		
Casos	19									0	0%	0		
Casos	20									0	0%	0		
Fin	20		Fin Opción							0	0%			

A la vez que se desarrollaban los proyectos (donde se agregaban casos funcionales y/o casos de reglas del negocio) se agregaban los casos de biblioteca, el resumen se mostraba en la página principal de la plantilla como se muestra en la **Figura 2.3**

Figura 2.3

Plantilla Resumen de Biblioteca

Biblioteca de Casos de Pruebas
 Nombre Aplicativo

Responsable de Pruebas: --- Seleccione ---
 Cantidad de casos: 500
 Sub-Modulos totales: 25
 Procentaje Completado: 0 %
 Fecha Inicio de Pruebas:

Modulos:

Nro.	Descripción	% Avance	Nro. de Casos	Sub-Modulos
1	Modulo1: Login	0%	100	5
2	Modulo2: Nueva Cotizacion	0%	100	5
3	Modulo3: Buscar Cotizacion	0%	100	5
4	Modulo4: Configuracion de Permisos	0%	100	5
5	Modulo5: Ayuda	0%	100	5
Total:		0 %	500	25

Módulos a Probar	Casos a Probar					
	Módulo 1	Módulo 2	Módulo 3	Módulo 4	Módulo 5	Módulo 6
	0	0	0	0	0	0

Rol Usuario x Proyecto		
Proyecto	ROL	Nombre Usuario

Casos Agregados por Proyecto.	
PXXXX	0
PYYYY	0
Biblioteca	0

El control de los casos agregados se realiza verificando los casos desarrollados para el proyecto y los que se agregaron como biblioteca comparando con una versión

anterior, luego mediante reunión programada el analista de control de calidad encargado explicaba el flujo del negocio y los casos que se desarrollaban en el proyecto, es así como luego se actualizaban los procesos en la plantilla de flujos como se muestra en la **Figura 2.4**

Figura 2.4

Flujos y sus casos asociados

PROCESO DEL NEGOCIO		PROYECTO BIBLIOTECA DE CASOS Flujo de Emisión												
No.	Flujo del Negocio	Biblioteca : RANSA			Biblioteca : VIAP OP			Biblioteca : SAS			Biblioteca : VIAP			
		Módulo	SubMódulo	Casos	Módulo	SubMódulo	Casos	Módulo	SubMódulo	Casos	Módulo	SubMódulo	Casos	
1	Ingreso de Solicitudes con Nuevos Productos	Digitación de Solicitudes VIAP	Digitación de Solicitudes VIAP											
2	Búsqueda de Solicitud Digitada	Búsqueda de Solicitudes	Búsqueda de Solicitudes VIAP											
3	Re Digitación de Solicitudes	Digitación de Solicitudes VIAP	Redigitación de Solicitudes VIAP											
4	Generación del Archivo de Texto para VIAP	Generar Archivo de Envío de Solicitudes	Generar Archivo de Envío de solicitudes VIAP											
5	Validación de Prima Excedente para Premium Vida Max y Vida Plus													
6	Rectificación de Solicitudes				Administración	Modificar								
7	Generación del Archivo de Texto para VIAP				Cargar Solicitudes	Generar Archivo de Envío de Solicitudes								
	Carga de Archivo RANSA en VIAP												Herramientas	Solicitudes
	Aprobar Solicitud y Convertir en Póliza													

APPS que intervienen a lo en determinado proceso del negocio.

En la **Figura 2.4** tenemos como ejemplo un flujo de emisión de una póliza, desde donde se origina una póliza de determinado producto, entonces cada hoja viene a ser un determinado producto del mismo tipo de seguro.

Es así como se deja una base de conocimiento para futuros analistas de Control de Calidad que ingresen al negocio y al cliente.

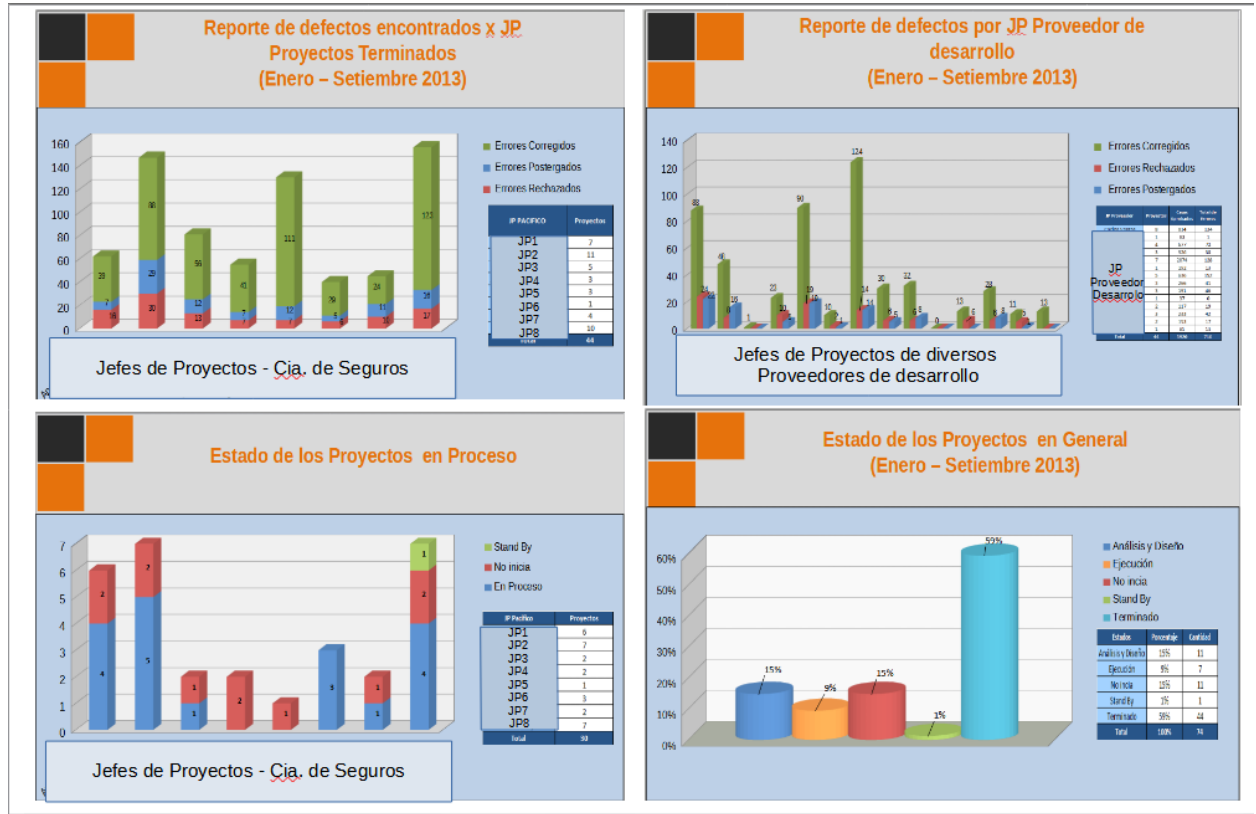
Adicionalmente a la gestión de la biblioteca, también fui encargado de emitir reportes gerenciales semanales al cliente que se muestran en la **Figura 2.5**

- Reporte de defectos por jefe de Proyecto del Cliente.
- Reporte de defectos por jefe de Proyecto del Proveedor de desarrollo.
- Reporte de los proyectos en proceso
- Reporte de los proyectos en general

Toda la información necesaria para emitir los reportes era recopilada a lo largo de la semana para ser enviada los lunes.

Figura 2.5

Reportes Gerenciales



El objetivo de estos reportes era demostrar que el trabajo realizado por el proveedor de certificación es de gran ayuda para el cliente en las revisiones, que proveedor de desarrollo necesitaba refuerzo en sus equipos, y el seguimiento de los distintos proyectos que como proveedor de certificación se estaba realizando.

Por otro lado luego ingresar a otro equipo de calidad en otra empresa proveedora de soluciones tecnológicas, y ante la problemática de demora en la emisión de los reportes de ejecución diario y semanal dentro del apoyo que se brinda al equipo de desarrollo se decidió por investigar más sobre el uso del Mantis BT, hallándose una opción para exportar una lista de incidencias encontradas según el proyecto con su respectivo detalle, esta lista era exportable para ser usado por las hojas de cálculo.

Con los conocimientos previos de programación y elaboración de macros se realizó una pequeña macro piloto para asociar las incidencias a los casos de la plantilla y al verificar que la distribución era correcta se realizó un estudio de varios proyectos y se

obtuvo como información promedio lo que se muestra en la **Tabla 2.1**, con esta información se justificó la implementación de las mejoras en una plantilla integrada.

Estas mejoras en el proceso de pruebas podrían generar algunos beneficios como

- Aumento de las horas efectivas de ejecución de pruebas.
- De contar con un producto estable, la ejecución de pruebas podría acabar en menor tiempo
- Al acabar las pruebas de los proyectos en menos tiempo se daría paso otros proyectos aumentando así la productividad del equipo.

Tabla 2.1

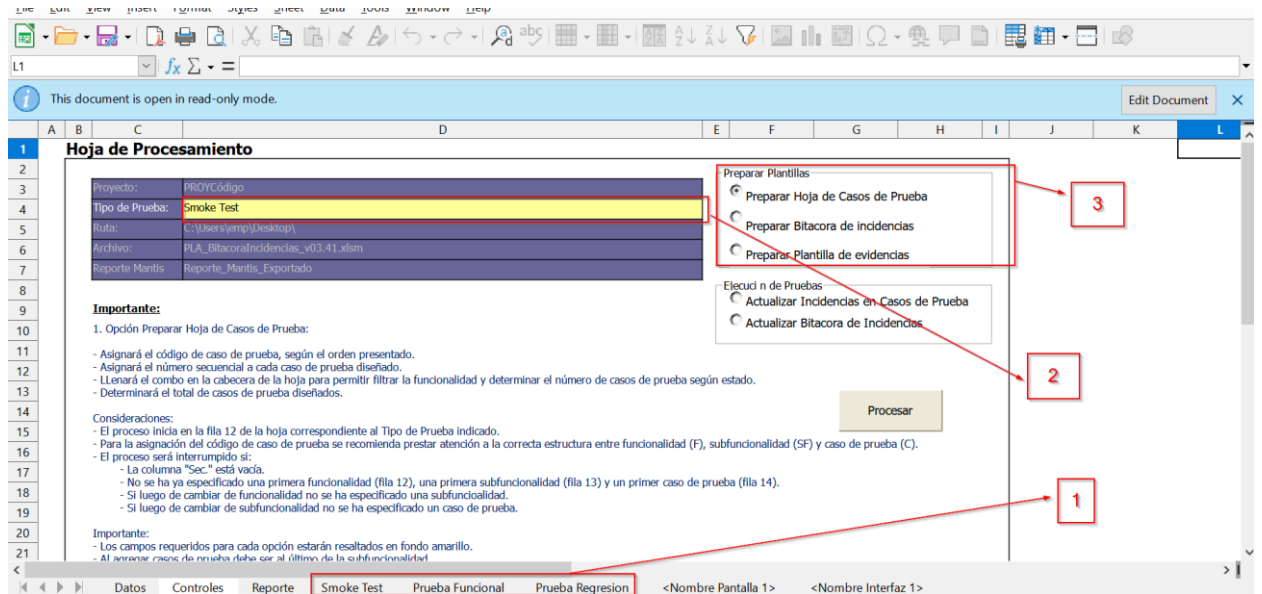
Justificación para la mejora de la plantilla y generador de reportes de ejecución de pruebas.

Resumen del diseño	
Tipo de prueba	Casos desarrollados
Pruebas Smoke	50
Pruebas Funcionales	200
Pruebas Regresión	150
Total de Casos	400
Casos a Ejecutar	
Casos ejecutados	
Ejecución por hora	10
Avg Incidencias encontrados por día	10
Avg Incidencias por Caso	3
Si el Total de Casos es 400:	
Con los bugs encontrados el total de casos a ejecutar	610
Casos por día	8 horas por día
Proyección x día	80
Proyección x Semana	400
Realidad	6 horas por día
Proyección x día	50
Proyección x Semana	250
	Duración de la ejecución en Semanas
Sin la automatización de Reportes	2.44
Realidad	7 horas por día
Proyección x día	75
Proyección x Semana	375
	Duración de la ejecución en Semanas
Con la automatización de reportes	1.626666666666667

En la figura 2.6 se presenta una versión de las plantillas generadas para la automatización de generación de reportes.

Figura 2.6

Presentación de la Plantilla



Inicialmente se creó una macro para importar la lista generada del Mantis BT con todos los bugs y mediante una macro creada, era generado un reporte preliminar, de acuerdo con los códigos del caso de Prueba generados por una macro ya existente, fue por esto por lo que se necesitó de una herramienta integrada para cumplir con todo el proceso de diseño y ejecución de pruebas para determinado proyecto.

Como se puede apreciar en la figura ítem “2” se escoge el tipo de prueba que se necesitaba codificar y en el paso o ítem “3” de esta misma figura y junto con el botón “procesar”, habiendo seleccionado previamente los datos del panel de la izquierda, realizamos la codificación como se puede visualizar en la siguiente **Figura 2.7**.

Figura 2.7

Codificación de casos de prueba

Diseño de Casos de Prueba - Prueba Funcional

Funcionalidad:

Todo	
Confirmar	0
No Confirmar	0
No Aplica	0
Pendiente o Sin Estado	13

Importante:

- Funcionalidad, Columna E = "F" - Columna C de 4 caracteres (Ejem.: CP01). Utilizado para llenar el combo.
- SubFuncionalidad, Columna E = "SF" - Columna C de 7 caracteres (Ejem.: CP01.01). En el caso de no existir duplicar
- Caso de Prueba, Columna E = "C" - Columna C de 10 caracteres (Ejem.: CP01.01.01)
- Pasos del Caso de Prueba, Secuencial, Columna E diferente de "F" o "SF" o "C" o "" (vacío)

(*) Celdas a considerarse en el diseño de los Casos de Prueba

Nro.	Clasificación	Código Caso de Prueba	Pasos a Seguir	Condición Inicial	Resultado Esperado	Estado
	CP01	F	RFI Se requiere modificar el formulario de Registro de Personas.			
	CP01.01	SF	Registro de Personas - SICAP			
	CP01.01.01	C	Registrar una persona sin cuenta bancaria - Persona Natural			
			1 Ingresar a Mantenimiento Generales > Personas.			
			2 Registrar una cuenta Bancaria			
	CP01.01.02	C	Registrar una persona con cuenta bancaria desde Menu Mantenimiento			
			1 Ingresar desde Mantenimiento de Personas			
			2 Registrar una cuenta Bancaria			
	CP01.01.03	C	Registrar una persona con cuenta bancaria desde el ingreso			
			1 Ingresar desde la Pantalla Inicial al Registro de Cuentas Bancarias.			
			2 Registrar una cuenta Bancaria			

Datos Controles Reporte Smoke Test Prueba Funcional Prueba Regresión Revisión MDI Reporte MDI - detalle Herramienta <Nombre Pantalla 1> <Nombre In

Con todos los casos desarrollados se procede a preparar los reportes de ejecución y generación de las plantillas de evidencias para el tipo de prueba que íbamos a ejecutar en la opción “Preparar plantilla de evidencia”, el cual contendría los casos desarrollados y previamente codificados con el espacio necesario para colocar las evidencias por cada caso ejecutado.

Cabe mencionar que al inicio las evidencias se entregaban sin alguna uniformidad en el diseño del documento.

Figura 2.8

Reporte de ejecución

Métrica de Calidad de Software en base al "Avance de los Casos de Prueba"

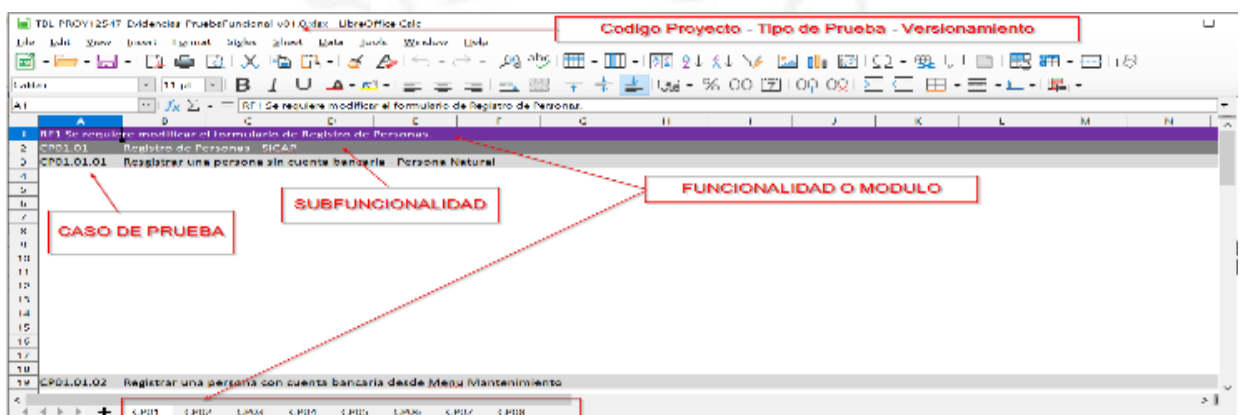
Funcionalidad	Diseño		Ejecución					Avance de Casos de Prueba (%)		
	# Casos de Prueba	Casos de Prueba (%)	Conforme	No Conforme	No Aplica	Pendiente	Esfuerzo del AQC %		# Casos de Prueba Ejecutados	# Casos de Prueba Ejecutados Conforme
Smoke Test	176	22.65 %	165	2	1	8	95.45 %	168	166	94.32 %
RF01 Modificaciones en el Mantenimiento de Campos de	7	3.98 %	7	-	-	-	100.00 %	7	7	100.00 %
RF02 Modificaciones en el Mantenimiento de Flujos	5	2.84 %	5	-	-	-	100.00 %	5	5	100.00 %
RF03 Lista de Solicitudes Pendientes de Aprobación	3	1.70 %	3	-	-	-	100.00 %	3	3	100.00 %
RF04 Modificación en el Mantenimiento de Tipo de	15	8.52 %	15	-	-	-	100.00 %	15	15	100.00 %
RF05 Mantenimiento de Excepciones de Flujo	8	4.55 %	8	-	-	-	100.00 %	8	8	100.00 %
RF06 Modificaciones en el Mantenimiento de Grupo de	6	3.41 %	6	-	-	-	100.00 %	6	6	100.00 %
RF07 Lista de Solicitudes Pendientes de Atención	4	2.27 %	4	-	-	-	100.00 %	4	4	100.00 %
RF08 Modificaciones en el Mantenimiento de Usuarios	1	0.57 %	1	-	-	-	100.00 %	1	1	100.00 %
RF09 Mantenimiento de Grupos de Tipos de Solicitud	6	3.41 %	6	-	-	-	100.00 %	6	6	100.00 %
RF10 Mantenimiento de Documentos	6	3.41 %	6	-	-	-	100.00 %	6	6	100.00 %
RF11 Mantenimiento de Destinatarios por Etapas	3	1.70 %	3	-	-	-	100.00 %	3	3	100.00 %
RF12 Modificaciones en Registro de Solicitudes	18	10.23 %	18	-	-	-	100.00 %	18	18	100.00 %
RF13 Modificaciones en Bandeja de Aprobación	3	1.70 %	2	1	-	-	100.00 %	3	2	66.67 %
RF14 Modificaciones en Asignación de Atención de	3	1.70 %	3	-	-	-	100.00 %	3	3	100.00 %
RF15 Modificaciones en Atención de Pendientes	6	3.41 %	6	-	-	-	100.00 %	6	6	100.00 %
RF16 Modificaciones en el Seguimiento de Solicitudes	4	2.27 %	4	-	-	-	100.00 %	4	4	100.00 %
RF17 Modificaciones en la Consulta de Mis Asignaciones	1	0.57 %	1	-	-	-	100.00 %	1	1	100.00 %
RF18 Modificaciones en la Consulta de Mis Solicitudes	2	1.14 %	2	-	-	-	100.00 %	2	2	100.00 %

Datos Controles Reporte Smoke Test Prueba Funcional Prueba Regresión MantenimientoCamposSolicitud AgregarNuevoFormato MantenimientoFlujos

Si se observa en la **Figura 2.8** en esta versión de la plantilla, se presenta un reporte de ejecución, pero la funcionalidad de la generación del reporte de ejecución se realizaría en una versión posterior.

Figura 2.9

Plantilla de Evidencias generadas para un proyecto en específico



Como se mencionó anteriormente, no se contaba con una uniformidad de plantilla de evidencias, con una versión posterior de la herramienta se construyó la opción que generaba dicha plantilla tal como se muestra en la **Figura 2.9**.

Figura 2.10

Reporte Exportado del Mantis BT



Proyecto	ID	Bitacora	Tipo de Prue	Funcionalid	Caso de Prue	Trazabilidad	Resumen	Tipo	Impacto	Prioridad	Estado	Fecha de eni	Actualizada	Asignada a	Informador
SSAC FASE 02	1827	1	Prueba Func	RF11 Manter	CP11.01.01	0	El formato d	Error	Bajo	Baja	En Proceso	20-02-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	1838	2	Prueba Func	RF11 Manter	CP11.01.01	0	Corregir info	Error	Bajo	Baja	En Proceso	23-02-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	1839	3	Prueba Func	RF11 Manter	CP11.01.01	0	Corregir el o	Error	Medio	Media	En Proceso	23-02-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	1840	4	Prueba Func	RF11 Manter	CP11.01.01	0	Corregir el o	Error	Medio	Media	En Proceso	23-02-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	1843	5	Prueba Func	RF11 Manter	CP11.01.01	0	Corregir nom	Error	Bajo	Baja	En Proceso	23-02-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	1850	6	Prueba Func	RF11 Manter	CP11.01.02	0	Al realizar la	Error	Medio	Media	En Proceso	23-02-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	1851	7	Prueba Func	RF11 Manter	CP11.01.02	0	Al realizar la	Error	Medio	Media	En Proceso	23-02-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	1854	8	Prueba Func	RF11 Manter	CP11.03.01	0	Actualizar d	Error	Medio	Media	En Proceso	23-02-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	2044	9	Prueba Func	RF23 Consult	CP23.02.01	0	Nombre de l	Error	Bajo	Baja	Por Despleg	10-03-2015	15-09-2015	jtorresn	Icalero
SSAC FASE 02	2051	10	Prueba Func	RF05 Manter	CP05.05.01	0	Opción "EXP	Error	Alto	Alta	En Proceso	10-03-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	2053	11	Prueba Func	RF05 Manter	CP05.02.01	0	No se muest	Error	Bajo	Baja	En Proceso	10-03-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	2078	12	Prueba Func	RF05 Manter	CP05.03.01	0	Nombre del	Error	Bajo	Baja	Por Despleg	12-03-2015	15-09-2015	maguillar	Icalero
SSAC FASE 02	2081	13	Prueba Func	RF05 Manter	CP05.02.01	0	No se m ue	Error	Medio	Media	Por Despleg	12-03-2015	15-09-2015	maguillar	Icalero
SSAC FASE 02	2082	14	Prueba Func	RF02 Modific	CP02.02.05	0	Se permite l	Error	Severo	Alta	Por Despleg	12-03-2015	14-09-2015	maguillar	jquispe
SSAC FASE 02	2085	15	Prueba Func	RF05 Manter	CP05.02.01	0	Al seleccion	Error	Bajo	Baja	En Proceso	12-03-2015	14-09-2015	maguillar	Icalero
SSAC FASE 02	2087	16	Prueba Func	RF05 Manter	CP05.02.01	0	El formato	Error	Alto	Alta	En Proceso	12-03-2015	14-09-2015	maguillar	Icalero

Como se había mencionado anteriormente, la emisión del reporte de ejecución, tanto diario como semanal, demandaba un tiempo que disminuía de manera considerable el tiempo efectivo de ejecución. Dado el problema y la implementación de la mejora, una vez obtenida la exportación de la lista de las incidencias publicadas en el MantisBT, que se muestra en la **Figura 2.10** se cargaban en la plantilla de la **Figura 2.6** y se ejecutaba ambos procesos “Actualizar incidencias en casos de prueba” y “Actualizar bitácora de incidencias” (la que se muestra en la **Figura 2.11**), con esto se lograba que el reporte o informe de ejecución se lograra en mucho menos tiempo del que se usaba normalmente.

Figura 2.11

Plantilla para la bitácora de incidencias

Reporte de Incidencias

1 A. Incidencias por Tipo e Impacto / B. Incidencias por Tipo y Prioridad

Tipo	A.Incidencias por Tipo e Impacto					B.Incidencias por Tipo y Prioridad					Total
	Severo	Alto	Medio	Bajo	(En Blanco)	Urgente	Alta	Media	Baja	(En Blanco)	
Error	-	-	-	-	-	-	-	-	-	-	-
Sugerencia	-	-	-	-	-	-	-	-	-	-	-
Hallazgo	-	-	-	-	-	-	-	-	-	-	-
Consulta	-	-	-	-	-	-	-	-	-	-	-
Documento	-	-	-	-	-	-	-	-	-	-	-
Total	-	-	-	-	-	-	-	-	-	-	-

2 Incidencias Tipo Error por Tipo de Prueba e Impacto

Tipo de Prueba	Incidencias Tipo Error por Módulo e Impacto				
	Severo	Alto	Medio	Bajo	Total
Smoke Test	-	-	-	-	-
Prueba Funcional	-	-	-	-	-
Prueba Regresion	-	-	-	-	-
Total	-	-	-	-	-

Como conclusión y aportes en esta asignación laboral, la implementación de la plantilla agilizó procesos de diseño, ejecución de pruebas disminuyendo los tiempos de elaboración de plantillas individuales, uniformizando la documentación por proyecto, la oportuna entrega de reportes de la ejecución diaria durante la jornada de trabajo hizo más efectiva y eficiente la comunicación con el equipo de desarrollo para la entrega de incidencias solucionadas.

Como parte de la capacidad de gestión siempre he disfrutado de realizar mejoras para el equipo y así puedan utilizarlas en beneficio de su aprendizaje y del desarrollo de las actividades diarias, en este caso en mi empleo actual al notar que las revisiones con los analistas de negocio requerimientos terminados y “testeados” tomaban más del tiempo programado, se vio la necesidad de crear un repositorio o “workspace” tal y como se muestra en la **Figura 2.12**, este “workspace” estaría organizado con diferentes colecciones por microservicio desarrollado, a su vez cada uno de estos “colecciones” tienen diferentes requests o llamadas según el tipo de operación, cada uno de estas llamadas a su vez, tienen un pre script que debe ejecutarse para que tome un token obtenido de un microservicio de login y así poder realizar la operación necesaria y al ejecutarse la operación o el llamado y al obtener la respuesta del servicio, también evalúe automáticamente el estado de la llamada si fue exitoso o no, la estructura y ciertos valores en la respuesta.

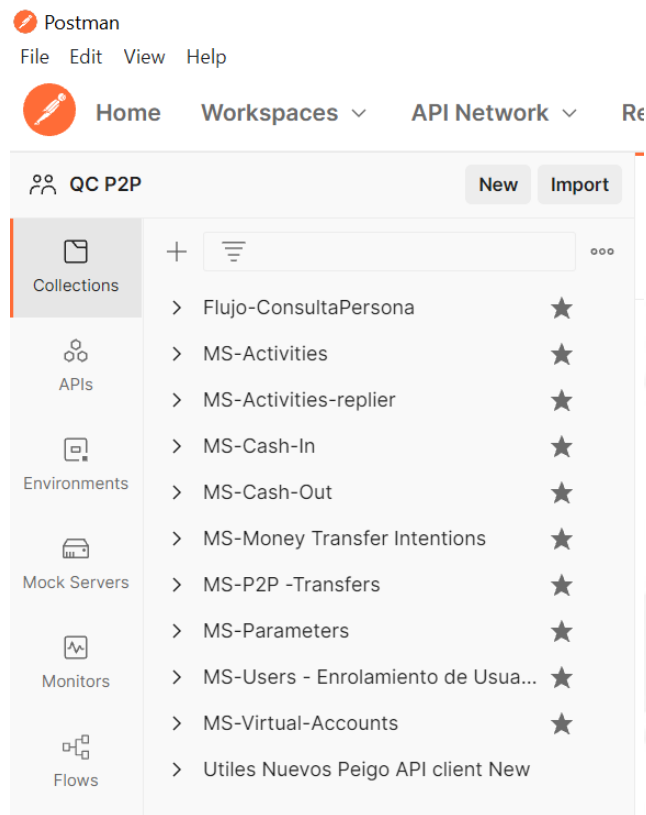
Ahora se pueden notar los beneficios de esta implementación, ya que el mantenimiento es sencillo al agregar nuevos servicios desarrollados o actualizar los

modificados por el desarrollo y sobre todo en el reúso para las pruebas de cada lanzamiento de nuevas versiones de la aplicación.



Figura 2.12

Inventario de Microservicios utilizados



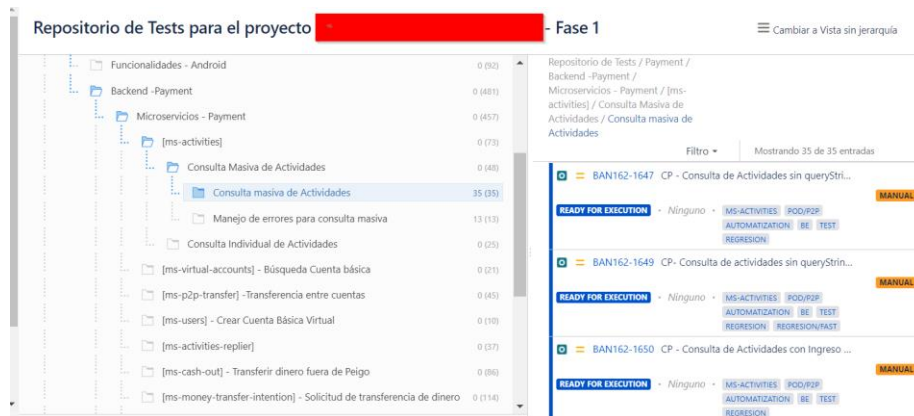
Este desarrollo de pruebas automatizadas para el equipo fue posible gracias al Inventario construido en una hoja de cálculo.

De acuerdo a la **Figura 2.13**, en la herramienta “JIRA”, también se vio conveniente organizar un repositorio por carpetas para cada microservicio, donde cada carpeta tendría los casos usados para el microservicio, esto permitiría el reúso de los casos de prueba, así como sus actualizaciones y si es requerido por la historia de usuario del sprint en curso.

Si bien mi experiencia en la gestión ha sido breve, ser miembro proactivo de los equipos donde he pertenecido me ha llevado a desarrollar mejoras e implementar soluciones en los procesos.

Figura 2.13

Inventario de casos de prueba por microservicio



Adicionalmente a la organización de los repositorios como su gestión continúa por el aumento de funcionalidades, esta organización de repositorios nos permite organizar y planificar las pruebas de regresión para los ambientes de UAT y producción para cada “release”.

3. APRENDIZAJE CONTINUO

Como profesionales en sistemas la tecnología creciente nos lleva a capacitarnos constantemente sino es a nivel formal mediante una entidad de capacitación entonces será a partir de capacitaciones internas mediante el empleador.

Muy temprano en mi carrera, con el objetivo de convertirme en un Administrador de Sistemas y de Redes inicié mi certificación en el track Microsoft Certified System Engineer, con la aprobación de algunos exámenes oficiales, pude lograr el Certificado Profesional de Microsoft, luego mucho tiempo después por el cambio de área laboral cuando me encontraba asignado a un cliente dedicado al rubro de telecomunicaciones como proyecto especial de Certificación y teniendo como instructor a nuestro líder del área de Calidad en el 2015 pudimos obtener la certificación ISTQB Certified Tester, Foundation Level para afianzar y reconfirmar todos mis conocimientos adquiridos en mi experiencia como Analista de Control de Calidad de Software, ya que hice uso de varias “técnicas de caja negra” usadas en la experiencia previa sin saber de la existencia de su nombre.

En el 2019 a sugerencia del empleador actual y con los conocimientos previos obtenidos en diversos talleres en entornos ágiles de trabajo logré una Certificación como Scrum Master, gracias a los conocimientos y la certificación pude ser parte de equipos dentro del marco “ágil” ideal para el negocio “retail” adicionalmente a estos conocimientos se me fue necesario aprender uso de herramientas para testing con móviles, tales como “mirrors” y el Android Studio para identificar los microservicios que eran llamados desde el app desarrollado por el equipo y evaluar el origen del defecto y reasignarlo al equipo de “backend” o “frontend” responsable del desarrollo, esto me llevó a ampliar mis conocimientos en pruebas de backend con la plataforma de educación “UDEMY”, obteniendo el certificación “Software testing de aplicaciones. Postman. Testing de API”. Adicionalmente dentro del Cliente se nos sugirió llevar una capacitación interna sobre Automatización de Pruebas tanto Web desktop como en equipos móviles (Android) con el framework CodeceptJS.

Posteriormente se fue necesario el aprendizaje de automatización en Selenium cuando se empezó un nuevo proyecto dentro del mismo cliente, automatización que no pude terminar por realizar un cambio de empleador, este curso ahora lo llevo por separado

a través de UDEMY por el hecho de no dejar los conocimientos inconclusos y ser Selenium uno de los frameworks más usados en el mercado.

Es para mi asignación actual en esta empresa Multinacional proveedor de TI para diversos clientes de la región que estoy usando los conocimientos adquiridos en el curso de Pruebas de API en postman, si bien el repositorio no fue algo solicitado por el cliente o los líderes del proyecto, creíamos conveniente desarrollarlo para tenerlo centralizado y que cualquier miembro del proyecto pueda usarlo para consultas en los servicios y como apoyo en las pruebas de frontend.

El desarrollo de este repositorio, así como sus inventarios ha servido también para realizar pruebas de regresión mucho más rápidas para poder entregar resultados dentro del tiempo requerido o incluso antes de la fecha estimada.

El contar con este repositorio e inventario también ha servido para proporcionar de manera oportuna la información requerida a los líderes (técnicos) y gestores del proyecto, tanto proveedor como cliente.

Si bien el negocio ahora no requiere y/o se está agregando miembros al equipo para las tareas de automatización, mis ganas de aprender y adquirir otros conocimientos es que me llevan al aprendizaje de Pruebas Automatizadas de backend desde un IDE con Rest Assured para poder ampliar conocimientos en pruebas que no son posibles realizarlos en otra herramienta por el requerimiento de licenciamiento adicional en Postman.

3.1 Tópicos de aprendizaje

Emprendimiento Tecnológico

Para empezar a comprender y ubicarnos dentro del amplio mundo del emprendimiento tecnológico, empezaremos con la definición de Startup como una empresa formada por individuos solos o en grupos, mejor dicho equipos destinados a crear nuevos productos de 0 como “una gran propuesta de valor” (Osterwalder A.2014) “en condiciones de incertidumbre extrema” (Ries 2011).

Tomando como base los ejemplos de startups, Rogers definiría a las startups como “plataforma”, un tipo de modelo de negocio, o “un negocio que crea valor al facilitar la interacción directa entre dos o más tipos distintos de clientes”.

Una Startup se diferencia de una Pyme por su juventud, Innovación que con una asociación con la tecnología correcta constituye una ventaja competitiva, sin precedentes en el mercado, por lo que pueden considerarse disruptivas, porque rompen patrones utilizados por otras empresas del mismo segmento

Los Startups también se diferencian de una Pyme porque el ámbito que se logra llega a ser global, tienen costos reducidos, porque no necesitan de un espacio físico destinado a su organización, se relacionan con la tecnología porque cuentan con ideas innovadoras. Buscando simplificar procesos y en cuanto a la financiación: buscan la inversión de terceros, personas que se involucran en la gestión empresarial, este aporte solo ocurre si la propuesta es realmente diferenciada a lo que el mercado ofrece y así tener la competitividad necesaria.

Lluís Soldevila nos ayuda a entender cómo ser parte de la transformación digital, y que es lo que se necesita.

- la **aptitud** como proceso innato, natural, aquel que necesitamos para ir adquiriendo conocimientos como parte de una evolución, pero a la vez nos enfrentamos a cosas que tal vez no entendamos y que no son tan triviales o básicas en el camino como tecnologías más avanzadas (IA).
- Para hacer frente a lo anterior, necesitamos de la **actitud**, con la correcta uno estará más dispuesto al cambio, motivación para adaptación y adquirir nuevas habilidades, esas que nos permitirán entender las nuevas tecnologías, las avanzadas, cosas más complejas.

A su vez la actitud correcta nos ayudará a resolver problemas, superar desafíos, porque rasgos como la determinación, tenacidad y resistencia se harán visibles e importantes.

- **Altitud** : Será determinada por nuestra actitud, por una mentalidad positiva, y que sería más importante que la capacidad o los conocimientos técnicos. De qué nos serviría esta capacidad técnica, si tenemos un problema y nos rendimos o nos hace dudar de nuestras mismas capacidades, es la actitud que pongamos para resolver el problema, y tener el problema resuelto nos dará la altitud.

De los Startups más conocidos, tenemos:

Airbnb: A diferencia de los grandes hoteles, este tipo de “startup” posibilita que personas en todo el mundo pongan a disponibilidad sus casas o sus inmuebles deshabitados, como alquiler por cortos periodos de tiempo para viajeros de estadias de corto tiempo. Este ofrecimiento lo hacen por medio de aplicaciones web, o los promocionan a través de redes sociales.

UBER: Similar al Airbnb, Uber te conecta con conductores privados para llevarte al lugar que tu indiques por medio del aplicativo, se ha convertido en la gran competencia de los taxis.

Spotify: Luego de que sitios y aplicativos de descarga ilegal de música estaban siendo intervenidos, llega Spotify que, mediante un pago mensual, los usuarios tienen acceso a una gran biblioteca de canciones.

Con la evolución de las redes móviles, ha sido disruptivo para que UBER, Spotify, puedan hacer uso de “Tecnologías Digitales” de la Industria 4.0, estas tecnologías hacen posible eliminación de intermediarios, por la manera en que la gente se conecta, la interacción con los clientes, manejo de grandes volúmenes de datos, una de las cosas más importantes sobre estas tecnologías digitales es que nos permiten experimentar, probar, algo que antes nadie se atrevía.

Las tecnologías digitales actúan en diferentes sectores en sus diferentes etapas de operación y producción, de manera diferente en su forma de implementación y articulación con otras áreas, sectores y organizaciones.

Las tecnologías digitales de la industria 4.0 son:

- Analítica avanzada e Inteligencia artificial.
- Internet de las Cosas (iot).
- Robótica Avanzada
- Servicios en la nube y plataformas digitales
- Cadena de bloques (Blockchain)
- Navegación autónoma y semiautónoma.
- Impresión 3D
- Realidad Virtual y realidad aumentada (CEPAL 2022)

Digitalización en el mundo

Si bien ya había empezado años antes; con la aparición del COVID 19 y el respectivo confinamiento, la transformación digital se ha acelerado, Si pensamos en el startups y como han usado las tecnologías, pensemos en el ejemplo del startup de “UBER”, como el proceso de pedir un taxi se convirtió en un proceso digital, desde solicitar que te recojan en una ubicación y asignar una ubicación destino, te calcula una tarifa y decides si lo tomas o no en ese momento, incluso si se posee una tarjeta de crédito asociada ya no es ni necesario tener dinero en efectivo, si nos ponemos a pensar hacia atrás en lo que era pedir taxi, habían 2 formas muy conocidas.

1. Salir a la calle, llamar al taxi, preguntarle el precio y si estamos de acuerdo tomamos el taxi.
2. Llamar por teléfono a una compañía de taxi, darle las indicaciones y acordar el servicio para el recojo.

Estos 2 procesos se digitalizaron gracias a expertos de distintas áreas y con la perspectiva externa de un consultor de negocio. pero para leer estas oportunidades, hay que estar muy pendientes de las necesidades, demandas del mercado, si se logra causar un impacto positivo en tu entorno y ser diferente, estaremos más cerca del éxito.

Para José Manuel González-Páramo combinar unas buenas regulaciones para la infraestructura digital (Internet) y lograr que la población acceda a una “*educación financiera mínimamente relevante*”, se convertiría en el escenario donde lo digital se convierta en una oportunidad de mejoras de condiciones de vida de la gente, permitiendo el crecimiento de la inclusión financiera rompiendo barreras geográficas para suministrar productos financieros con cumplimiento regulatorio a bajo costo transformando la necesidad en una realidad.

Conociendo que son los startups, el uso de las tecnologías y algunos conceptos de transformación digital, conozcamos algunas metodologías o marcos de trabajo poder llevar a cabo un “startup”

a) Design Thinking:

Según Ana Svalina así como otros autores definen “Design Thinking” como una manera innovadora de pensar y resolver problemas enfocándose en el usuario. A su vez, el desarrollo de la teoría de “Design Thinking” depende del pensamiento humano en el campo de ciencias cognitivas, inteligencia artificial y neurociencia, por lo tanto “Design Thinking es integral, interdisciplinario y difícil de definir sin ambigüedad y simple, es por eso por lo que no hay una definición final de “Design Thinking”. El término “Design Thinking” es usualmente usado como un enfoque único de resolver problemas de una manera innovadora. (Svalina et al. 2022 p.447)

Amie Cieminski define también “Design Thinking” como un proceso colaborativo e iterativo en el cual los diseñadores buscan entender al usuario, asumir desafíos y redefinir problemas en un intento por identificar estrategias alternativas. (Cieminski 2022 p.6-12))

Es en el intento de identificar las estrategias alternativas que se encuentra la coincidencia en lo que es tecnológicamente factible, minimizando el riesgo y aumentando las probabilidades de éxito.

- Este “Pensamiento de diseño” es todo un proceso creativo participativo en donde colaboran los empleados, clientes, proveedores y profesionales de distintas disciplinas y para desarrollar todo este proceso, las personas deben ser colaborativas, observadoras, empáticas, les gusta experimentar, optimistas, pacientes, integradores y con visión global y personalizan.
- En una sociedad de constante cambio y se están reinventando a sí mismas, es en la reinvención como interviene el “Design Thinking” para generar nuevas ideas y esquemas de funcionamiento
- Cuando nos referimos a Pensamiento de diseño no hablamos del acto de diseñar, el pensamiento de diseño es el proceso que se ha seguido para llegar al resultado diseño concreto y tangible.
- Pero cómo desarrollar este pensamiento de diseño, ¿está nuestra mente preparada para la innovación?, mientras nos quedemos en nuestra zona de confort mental, será imposible, debemos arriesgarnos y salir a un territorio

desconocido, mientras más alejemos de nuestra zona de confort, más aprendizaje y creatividad desarrollaremos, y es ahí donde surgen grandes retos y las ideas brillantes.

- De las metodologías usadas por los diseñadores son el Modelo de Stuart Pugh, cuyo modelo consta de 11 etapas en su versión original hasta el retiro del producto del mercado, existe otro método llamado Scamper que ayuda en la generación de ideas, Mapas conceptuales de Novak, Mapas mentales de Buzan, Storytelling, modelo de Bruno Munari, todas las metodologías revisadas se resumen en una serie de etapas: Comprender, observar, definir, idear, prototipar, testear, implementar y aprender.
- Si los resultados exceden las expectativas iniciales o la sobrepasan es que el proceso de pensamiento de diseño ha sido exitoso, para eso debemos desarrollar una metodología que sea atractiva y dinámica para que lo podamos usar como estrategia de innovación en nuestros negocios y procesos actuales.

b) Lean Startup

Switek, Drelichowski, concluyen que Lean Startup es un nuevo método de entender la organización, aunque tiene una relación floja con el método de administración Lean usando algunas de sus herramientas, el propósito de ambos es evitar el desperdicio. Lean Startup, sin embargo, trabaja en el área de extrema incertidumbre, cuando los requerimientos de los clientes no son conocidos, proponiendo solo lo que el cliente puede aceptar. (Switek, Drelichowski 2018 p. 30)

Si separamos las palabras del término Lean Startup, obtendremos que Lean significa Magro, algo que no genera valor y startup pues ya lo hemos definido anteriormente, como la puesta en marcha de un negocio, ésta puesta en marcha es realizada en 3 pasos: Construir, Medir, Aprender.

Del primer paso denominado como “Construir” el objetivo es llevar la idea a un producto material, obtendremos como primer producto, un producto denominado como Producto Mínimo Viable (PMV) (MVP como se suele escuchar actualmente) construido en base a características prioritarias y puesto

a disposición de usuarios y clientes para obtener las primeras impresiones, (feedbacks).

Una vez entregado el producto a usuarios potenciales empieza la etapa de la “Medición” para eso debemos contar con herramientas de medición, para recoger las primeras impresiones del producto. Ante un “PMV”, podemos usar “métricas piratas” como adquisición, activación, retención, referencia, ingresos.

Obtenido los datos de las métricas aplicadas, empieza el 3er paso “Aprender” en el análisis de los resultados recogidos, a partir de aquí se vuelve a iniciar el ciclo para elaborar el producto final, es así como vamos descubriendo necesidades, funcionales, sociales y emocionales hasta llegar a todo un público objetivo.

Las principales ventajas al aplicar o al usar la metodología “Lean Startup” son:

- Eliminamos riesgos
- Se eliminan procesos inútiles
- Cumple con las necesidades
- Estructura las ideas
- Reduce fracasos

c) **Ágil Scrum**

La guía SBOK (Guía para el cuerpo del conocimiento de Scrum) dice que “Scrum es uno de los métodos ágiles más populares. Es un “framework” adaptable, iterativo rápido, flexible y eficaz, diseñado para ofrecer un valor considerable de forma rápida a lo largo del proyecto.” (Sathpaty T. Guía SBOK, p 2).

De la explicación gráfica de Sathpaty en la Guía SBOK y con cierta experiencia en el marco ágil, se podría explicar el proceso de la siguiente forma:

- En una reunión de stakeholders, se crea la visión del proyecto.
- Existe un “product owner” quien recibe de los stakeholders del proyecto los requerimientos (sin refinar) y establece un “product backlog”, un scrum máster que se encarga de crear el ambiente de trabajo para todos los miembros del equipo que trabajara en el desarrollo del proyecto resolviendo cualquier problema, desbloqueando o eliminando cualquier impedimento para poder lograr los objetivos de cada iteración denominada como “sprint”
- Cada Iteración o Sprint tendrá como entrada una lista de requerimientos llamado Historias de usuario, las cuales son previamente escritas y estabilizadas por el “product owner”, pero deben ser refinadas con todo el equipo para que se tenga conocimiento de que es lo que se va a hacer y prever cualquier problema futuro.
- Al inicio de cada sprint, con los requerimientos ya refinados, se realiza el “sprint planning”, donde se hace un “pesaje” de cada historia de usuario del sprint backlog y se establecen las prioridades.
- Durante el sprint cada día a través de las ceremonias “daily stand up”, se conversa entre todos los miembros del equipo acerca del progreso, trabajo realizado, lo que se realizará, haciendo los movimientos en un tablero “Scrum” y sobre todo cualquier impedimento que se pueda presentar en el desarrollo de los requerimientos, para que el Scrum Master tome nota y haga las coordinaciones necesarias.
- Finalizado el sprint, se llevan a cabo las siguientes ceremonias, el “sprint review”, donde se presenta todo lo terminado con un balance de lo planeado versus lo logrado ante el product owner y otros stakeholders con el objetivo de recibir el feedback. Otra ceremonia al final del sprint es “sprint retrospective”, dónde el equipo scrum habla de lo que se hizo bien, se hizo mal y lo que puede mejorar, además también tiene como entrada el “feedback” recibido del “sprint review”.
- Mejores prácticas para emprendimientos tecnológicos
- No hay un único manual que aplique para todos los casos de éxito, sin embargo, de datos recogidos de éxito y fracasos se han podido recoger las

siguientes mejores prácticas para mejorar los aspectos principales de un proyecto.

Para llevar el proyecto de la mejor manera dentro del marco scrum hay una serie de recomendaciones, además del conocimiento técnico también es necesario ciertas habilidades blandas para que podamos desarrollar los siguiente:

- Conoce tu producto mejor que nadie (y saber explicarlo a otros)
- Rodéate de las mejores personas que puedas
- Ábrete al feedback
- Analiza fríamente el perfil de tus inversores
- Networking, tu mandamiento.
- No apurar los tiempos
- Que hablen de ti
- Acércate a tus clientes.

d) Business Model Canvas (BMV)

El BMV es una herramienta de administración estratégica y emprendimiento. Te permite describir, diseñar, desafiar, inventar, hacer cambios en tu modelo de negocio.

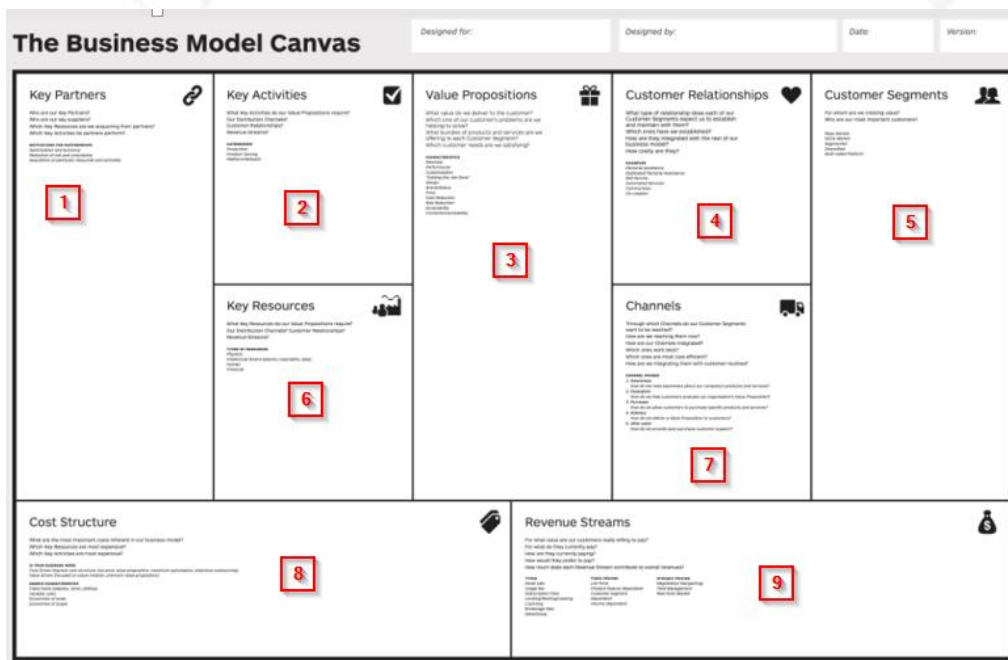
Como indica Avdiji en su artículo de investigación, en el panorama de negocio caracterizado por la complejidad e incertidumbre, donde los modelos de negocio demandan la coordinación de un número creciente de stakeholders, está la necesidad de una herramienta para entender el mapa y la lógica del negocio. El “BMV” define modelos de negocio teniendo 9 componentes presentándolos en una plantilla visual para facilitar la ideación, elaboración y evaluación de ideas de modelos de negocio (Avdiji et. al. 2020 p.698) con el objetivo de que se convierta en el plan del negocio.

En la **Figura 3.1** mostramos el Business Model Canvas (BMV) con los componentes que se listan a continuación, tanto en español como su equivalente en inglés:

1. Socios Estratégicos / Key Partners.
2. Actividades fundamentales / Key Activities.
3. Propuesta de valor / Value Propositions.
4. Relación con los clientes / Customer Relationships.
5. Segmentos de mercado / Customer Segments
6. Recursos / Key Resources
7. Canales de distribución / Channels
8. Estructura de costes / Cost Structure
9. Ingresos / Revenue Streams

Figura 3.1

Business Model Canvas



Nota : Value proposition design(Osterwalder A. 2020 p. XV) – www.strategyzer.com

Es importante llenar todos los espacios no solo por una persona sino por varias de diferentes especialidades, puedes tener este tablero en físico en un gran espacio de trabajo con todo el equipo y/o manejarlo con plataformas tecnológica como miró, para llevarlo digitalmente con la capacidad de ser modificado en línea de manera compartida.

Ahora que conocemos varios tópicos acerca de Emprendimiento Tecnológico y cómo empezar, y si no tenemos el “expertise” o conocimiento técnico para poder

desarrollar un software capaz que conozca los estándares para ser un software seguro y confiable, podremos solicitar el apoyo de un analista programador que cuente con el conocimiento necesario.

Programación Defensiva

Lo primero que debemos tener en cuenta es no asumir nada durante la programación, lo cual significa que el código esté preparado para recibir los datos que se deben recibir y para recibir lo inesperado. Si bien dentro de los marcos ágiles de trabajo, en ciertas historias de usuario se definen estos escenarios (recibir lo inesperado) dentro del alcance del requerimiento, el deber y/o la visión del analista de control de calidad aparte de certificar que se cumpla el alcance, es proponer más escenarios que dentro de lo posible se puedan replicar en la realidad y a su vez también descartar escenarios donde la interfaz te restringe el ingreso de ciertos datos.

Dentro de los datos esperados debemos codificar de acuerdo con el tipo de dato, por ejemplo, si son valores numéricos que estén dentro de lo tolerado, si son valores alfanuméricos o cadenas de caracteres que sean válidos para el propósito previsto.

Para los valores no deseados, rechazarlos al momento de la entrada, incluso si la interfaz de usuario no lo permite, estos ingresos no deseados deben ser controlados o rechazados incluso desde los APIs invocados desde la aplicación, para que no los deje seguir en el código, ya que puede atacar el sistema posteriormente con intentos de desbordamientos de búfer, inyección de comandos SQL, código HTML o XML inyectado, desbordamiento de enteros, datos pasados a las llamadas del sistema.

Ante las entradas incorrectas hay varias maneras de manejarlas, por ejemplo:

- Técnicas de manejo de errores
- Registrar un mensaje de advertencia de un archivo
- Devolver un código de error
- Llame a una rutina/objeto de tratamiento de errores
- Mostrar un mensaje de error siempre que se encuentre el error

En las siguientes sería difícil un seguimiento que se está ingresando incorrectamente, pero serviría para confundir al atacante.

- Devolver un valor neutro

- Devolver la misma respuesta que la vez anterior

Durante la programación usamos varias técnicas para ir evaluando la funcionalidad y eficiencia de nuestro código, como son las aserciones y excepciones.

a) Aserciones:

Si buscamos el significado de una aserción se nos dirá que es una afirmación, aseveración, es una expresión que damos algo por cierto, entonces si vamos por el lado de la programación entonces se dice que una aserción es una expresión condicional que es asumida siempre verdadera, la falla en una aserción indica un error del programador (Becker P.1999 p. 267), dichas fallas siempre deben resultar en una excepción no marcada, si se recibió algo inesperado entonces la aserción alertará de lo sucedido. En las aserciones el código se comprueba a sí mismo y si está funcionando a lo esperado, generalmente es silenciosa y dejará seguir ejecutando el código.

Algunos lenguajes tienen soporte incorporado para utilizar aserciones, las aserciones son fáciles de escribir, pero también deben ser fáciles de remover y no deben afectar el rendimiento de la aplicación. por ejemplo en el lenguaje C++ no se proporcionan mensajes de texto, pero se puede realizar una función para que se emita el texto correspondiente.

Las aserciones son utilizadas en programas grandes, complicados y de alta fiabilidad y permite al programador depurar y recodificar para que no se produzca el error nuevamente, debemos usar para documentar suposiciones hechas en el código y para eliminar condiciones inesperadas.

Pautas para usar aserciones:

- No sobre usar aserciones lo que significa verificar todo lo que se crea que es verdadero.
- Si el código funciona sin una aserción entonces no se debe introducirlo.
- Si el código funciona y la aserción falla entonces remueva la aserción.
- Verifique duplicidad de código al invocar aserciones.
- Una aserción debe ser una pieza código de fácil extracción

- No deben afectar la “performance” en producción.

b) Excepciones

Una excepción es “cuando ocurre un error de alguna clase o un evento inesperado durante la ejecución de un programa”, ocasionalmente algo puede ocurrir mal, ya sea porque las cosas cambian o no lo teníamos previsto en ese entonces, en una aplicación lo que puede ocurrir es que el software falle y se detenga toda la operación, pero algo ya ha sido ejecutado hasta cierto punto.

Entonces es parte de la programación defensiva que existan estas rutinas de código, que sirven para que se pueda pasar errores o eventos excepcionales al código que lo llamó, o separar claramente lo que funciona de lo que no podría funcionar si se encuentra con algo inesperado. Estas rutinas de código hacen que la codificación y el entendimiento sea más fácil.

Estas excepciones se pueden manejar como:

- Detección preventiva de fallas.
- Detección retrospectiva de fallas. (Sommerville I. 2002 p. 404-405)

La estructura básica de una excepción es que use “throw” para lanzar un objeto de excepción, luego el código en un módulo de jerarquía de llamada captar ese objeto que fue lanzado con un “try catch”

El empleo de excepciones varía de acuerdo al lenguaje usado pero habría que tomar las siguientes consideraciones:

- Si la rutina que llama es responsable para probar la condición antes de la llamada la excepción debe ser “unchecked”
- Buscar todos los que llamen a rutinas para modificarlos al uso de la excepción.
- Si hay numerosos llamados, hágalo de forma manual modificando ciertas partes del código como creando nuevos métodos que llamen a la excepción y a su vez que el código actual llame al nuevo método, cada vez que se realice esto, debe compilar y probar para verificar que siga funcionando, luego de esto podrá eliminar el antiguo código.

- Tener en cuenta que para usar las excepciones deben ser para comportamientos inesperados o excepcionales
- No deben ser usados como sustitutos de pruebas condicionales.
- El uso de excepciones, lineamientos según el lenguaje utilizado en una corporación debe ser estandarizado.
- Someta la aplicación a pruebas de regresión luego de una refactorización o una revisión de código comparando el antiguo y el nuevo código para disminuir riesgos y asegurar funcionalidades existentes.

Debemos determinar cuánta programación defensiva debemos dejar en el código de desarrollo que irá a producción.

- Dejando el código que verifica errores importantes.
- Elimine el código que comprueba errores triviales.
- Quite el código que resulta en fallas fuertes.
- Deje el código que ayuda al programa a fallar con gracia.
- Registre errores para su personal de soporte técnico.
- Asegúrate de que los mensajes de error que dejas son amigables.

Cabe agregar que la programación es susceptible al error humano, para la programación defensiva no es inmune a los defectos (programar es susceptible al “error humano”), incluso código adicional añade complejidad al software, hasta puede volverlo pesado y lento si se comprueba datos pasados como parámetro de cada manera posible en cada lugar posible. Por esto debemos establecer nuestras prioridades en programación defensiva y así sabremos en qué módulos necesitamos colocar programación defensiva.

Después de todo lo explicado y revisado en esta sección, nos impulsa a empezar con un emprendimiento, ahora que ya tenemos idea de cómo empezar con Design Thinking, para ir con Lean Startup y entregar un “PMV”, tras el feedback que recibamos, podemos empezar a construir un backlog de requerimientos y llevar un plan de trabajo dentro del marco Scrum para desarrollar un producto más completo, al usar scrum, debemos asegurarnos de que el trabajo de los programadores cumplan con estándares de OWASP y si en algún momento debemos manejar medios de pago del usuario, será necesario que estas características que conozcan el “PCI Compliance”

4. CONDUCTA ÉTICA

A lo largo de mi experiencia laboral veo, pienso y puedo afirmar que siempre me he podido manejar dentro de lo estipulado en el código de ética y conducta profesional, incluso de aquellas empresas donde no se me requirió firmar cláusulas que involucren este código de ética.

Como Analista de Control de Calidad de software y según el punto 1.3 del Código de ética y conducta profesional de ACM “Ser honesto y Confiable” es parte de nuestro trabajo sobre todo cuando se menciona “Un profesional de la Informática debe ser transparente y proporcionar una información completa de todas las capacidades del sistema”, porque es en nuestros informes de estatus y cierre de pruebas, debemos mencionar todo el progreso real y como se encuentra el activo digital dentro del proceso de certificación o pruebas de control de calidad, de nada nos sirve, obviar bloqueantes que impidan el desarrollo de nuestro trabajo o proporcionar evidencias que no correspondan al proceso actual que estamos trabajando, porque cualquier problema posterior nos dejaría con poca credibilidad para procesos posteriores.

Como analista de Control de Calidad, y de acuerdo al proyecto se nos proporciona información sensible y confidencial para poder realizar nuestro trabajo para que se asemeje más a un entorno productivo, Los puntos “Respetar la confidencialidad” y “Respetar la privacidad” del código de ética y conducta profesional de ACM, además de contratos y acuerdos de confidencialidad firmados dentro del contrato laboral, nos exigen cumplimiento de estos puntos. Cuando se menciona que “...Los períodos de retención y eliminación de esta información deben ser claramente definidos, cumplidos y comunicados a los interesados. La información personal recopilada para un propósito específico no debe ser utilizada para otros fines sin el consentimiento de la persona...”, dentro del contexto de pruebas en ambiente productivo, los datos de los usuarios como cédulas o documentos de identidad, números de tarjeta de crédito o débito, contraseñas. solo son usados cuando se exige este ciclo de pruebas previos al lanzamiento de una nueva versión, finalizado este proceso, los datos usados en las pruebas de micros son borrados para no exponer los datos sensibles del usuario.

Dentro del proceso de Control de Calidad que realizamos para el APP de la Billetera electrónica se debe cumplir con todo un proceso de documentación de los artefactos que se usan durante el análisis, diseño, ejecución y cierre de las pruebas a través del JIRA, una herramienta bastante utilizada en la administración de proyectos, donde se registra desde los requerimientos llamados “Historias de Usuario” dentro del marco SCRUM hasta el proceso de Calidad y aceptación de Pruebas del Analista Funcional o Analista de Negocio (“BA”)

Cuando me desempeñaba como Analista de Control de Calidad dentro del sector Retail, con todos los conocimientos adquiridos sobre el Portal Web y su APP como integración con otras herramientas que administraban su configuración y contenido, las tablas involucradas en la Base de datos y su conexión con otros sistemas o aplicativos comerciales. tuve la tarea de liderar y formar nuevos analistas para que puedan realizar las tareas que yo realizaba anteriormente y puedan conocer del sistema, a lo largo de los ciclos de entrega de software a producción se les iba guiando y explicando acerca de cómo ocurrían dichas entregas de software. Bajo la herramienta de comunicación Microsoft Teams solicité estar incluido en cada grupo de conversación por requerimiento, y así siempre me mantenía informado de las modificaciones y las fechas de entrega e incluso en las pruebas con el usuario para tomar nota de todos los acuerdos y otras consideraciones, entonces con toda esta información y del nuevos analistas presentes en cada grupo se les podía capacitar en el funcionamiento y la casuística que podría presentarse en los nuevos requerimientos.

5. LECCIONES APRENDIDAS

En la primera etapa de mi experiencia profesional a través de mis líderes de equipo siempre tuve 2 expresiones que generaron una gran reflexión en mí y que lo llevo presente hasta ahora y cada vez que tengo la oportunidad se lo digo a las personas que me toca trabajar o formar dentro de mis equipos.

- No siempre el bien justifica los medios: Podemos vincular esta expresión dentro del código de ética y conducta profesional, por más que la acción realizada sea bien intencionada, puede causar daño, como por ejemplo exponer una contraseña de un usuario o de tu propio usuario al momento de la revisión de un equipo o computadora que necesita la instalación de software adicional no autorizado solicitado por el usuario para su conveniencia en sus labores diarias.
- Para ser bueno no es suficiente con serlo sino parecerlo: Si bien cierta habilidad obtenida por la experiencia también es importante llevar un comportamiento y demostrarlo a través de actitudes y acciones hacia los usuarios con cautela en las expresiones que vayamos a manifestar, si vamos a un plano más superficial, también es importante la presencia y pulcritud en la apariencia diaria.
- Por último mencioné que cuando inicié mis actividades como Analista de Control de Calidad de Software, lo tuve bastante presente y fue así cómo me convertí en un analista requerido en diversos proyectos.

Durante mi etapa de analista de control de calidad he tenido la oportunidad de aplicar varios conceptos acerca de la información, por ejemplo:

- De acuerdo con el principio “disponibilidad” de la Información y que esta tiene valor si es entregada en el momento correcto, nuestros reportes de avance de ejecución de pruebas, reportes de defectos, sólo tenía valor para el jefe de proyecto por el lado del cliente si era entregado en cierto momento del día, incluso si encontramos un defecto cuando el sistema o aplicativo ya estaba listo por desplegarse en producción, eso podría retrasar la entrega.

- también podemos mencionar los reportes gerenciales debían ser entregados cada lunes de cada semana al inicio del día como compromiso de la gerencia del proveedor de tecnología, solo así tenían valor porque era en ese momento en que se le daba la importancia debida, si por a o b era entregada pasado ese momento del día, ya no era revisado, ya no había “feedback” y se “perdía” entre todo el masivo de correos que el jefe de proyectos podría recibir.
- Mencionando otro principio de la información como “Integridad”, la entrega de reportes, deben contener información exacta para que se puedan tomar decisiones acertadas, incluso los defectos encontrados deben mantener una integridad para reportarlo, y así la persona encargada de su corrección pueda entender con todo el reporte completo, no era solo suficiente publicar el mensaje o la pantalla capturada del error sino, debíamos mostrar como se pudo obtener el defecto y si era posible también replicarlo para confirmar su reproducción.
- Bajo el concepto agilidad, cualquier dependencia en cuestión conocimiento, implementación y desarrollos, deben ser superadas y si es posible adquirir el “know how” para poder eliminar la dependencia, y así obtendremos agilidad, si bien en parte es responsabilidad del rol “Scrum Master”, si está a nuestro alcance eliminar la dependencia, será en nuestro propio beneficio y aprendizaje de autogestión y adquirir más conocimientos.

Finalmente dentro de lo experimentado en proyectos con metodología “ágil”, aún sigo observando la siguiente problemática

- Se continúa manejando la etapa de control de calidad dentro de cada sprint como una mini cascada, es decir siempre esperamos la finalización de la construcción del código para el inicio del “testing”.
- Parte de la agilidad es escribir escenarios o casos de prueba antes de que el código sea construido, si bien una “Historia de usuario”, contiene los escenarios principales del requerimiento a desarrollar, la tarea del analista de control de calidad es encontrar más escenarios posibles a los ya planteados durante el refinamiento como etapa previa al sprint.

- Hallar el número de analistas de control de calidad que deben ser incluido en el equipo scrum por cada célula y por “x” número de “Historias de usuario” incluidos en el sprint, para contar con tiempo suficiente para escribir los escenarios de prueba antes de que el código sea escrito. Definamos célula como equipo de trabajo dedicado a desarrollar cierto grupo de funcionalidades agrupadas por tipo, ejemplo en una aplicación de billetera digital destinamos una célula para todo lo que se refiere a funcionalidades de las transacciones bancarias, otra célula que se dedica a trabajar funcionalidades de la cuenta, desde el ingreso al sistema y actualizaciones de datos del usuario y otra célula que se dedica a ver funcionalidades de fidelización del usuario.
- Los requerimientos de los usuarios / stakeholder, varían con el tiempo por diversos motivos, uno de ellos es tener una funcionalidad disponible en el mercado para cierta fecha ya sea porque la competencia está planeando entregar algo similar, esto podría impactar en la capacidad del equipo scrum para entregar cierto número de historias terminadas para el “sprint review”, esto a su vez impacta en el compromiso del equipo para:
 - Priorizar en las tareas del backlog del sprint para entregar el compromiso acordado.
 - Ante la sobre exigencia por parte del cliente en la entrega de historias, las reuniones de refinamiento podrían ser impactadas en la asistencia o atención para intervenir, hacer consultas (los equipos scrum son diversos en roles, cada uno podría aportar algo al refinamiento del requerimiento a desarrollar).

6. GLOSARIO DE TÉRMINOS

- **Android:** Plataforma de desarrollo libre para aplicaciones móviles y a su vez sistema operativo para el tipo de aplicaciones (Gironés J. 2018)
- **API:** Es la abreviatura de Application Programming Interface, “Son traductores cuya función es conectar sistemas, software y aplicaciones”. (Cardozo L. ZENVIA. 2021)
- **Bug:** Para el ámbito del software ISTQB define en su syllabus del nivel “Foundation”, la palabra “bug” como defecto o falta cuando cierta parte del código es ejecutado, código que a su vez ha sido escrito y deducido como “error humano” (ISTQB CTFL-2018)
- **BDD:** Definimos BDD como una estrategia de desarrollo orientada a comportamiento como lo indica sus siglas en inglés “Behavior Driven Development” y definido en el syllabus” de la extensión “Ágil” de ISTQB como “allows a developer to focus on testing the code based on the expected behavior of the software. Because the tests are based on the exhibited behavior of the software, the tests are generally easier for other team members and stakeholders to understand “[Permite al desarrollador concentrarse en probar el código basado en el comportamiento esperado del software. Porque las pruebas están basadas en el comportamiento exhibido del software, las pruebas son generalmente más fáciles de entender para otros miembros del equipo y stakeholders] (ISTQB CTFL-AT - 2014)
- **Bug Tracker:** Definido el término “Bug” anteriormente entonces nos referimos como “bug tracker” a una herramienta de software para hacer el seguimiento de los defectos encontrados en el ejercicio del control de calidad de software, sus diferentes estatus quién es el tenor del bug en el momento de una consulta.
- **Codeceptjs:** Is the testing framework open source and MIT licensed [Es un marco de trabajo licenciado de código abierto del Instituto Tecnológico de Massachusetts para automatizar pruebas] (*CodeceptJS*, n.d.)

- **Collection:** Dentro de la herramienta Postman (útil para pruebas de APIs), las “collections” vienen a ser grupos de llamadas a los servicios/microservicios/endpoints, con sus distintas operaciones como GET, POST, DELETE, PATCH.
- **FIX:** Arreglo o solución de una incidencia o bug.
- **Framework:** La palabra framework significa “marco de trabajo” que tiene definida una estructura para seguir un objetivo específico, en programación, sería el conjunto de archivos clasificados en carpetas que definen reglas para seguir buenas prácticas de codificación que mejor se acomodan a una organización y sus lineamientos de acuerdo a las herramientas que ofrece.
- **Frontend:** En la página web o aplicación es la forma como se muestran los datos, es decir el diseño.
- **Gherkin:** Definido como un lenguaje simple que se acerca al natural, también es definido de manera técnica como un lenguaje específico de dominio o Domain Specific Language (DSL). Es de gran ayuda a las personas que intervienen en un proyecto de desarrollo de software, desde el stakeholder hasta el tester, y sirve para escribir necesidades, documentación y pruebas. (Morales M. 2016)
- **IDE:** “Es el acrónimo del término inglés *Integrated Development Environment* o, lo que es lo mismo, **Entorno de Desarrollo Integrado**. Es el escenario digital utilizado en programación para desarrollar aplicaciones, juegos... Es imprescindible tanto en el ámbito del Desarrollo de Aplicaciones Web (DAW) como en el Desarrollo de Aplicaciones Multiplataforma (DAM). Hace que la tarea del programador sea más sencilla, gracias a las herramientas que tiene incorporadas, como compiladores, depuradores o bibliotecas, y esto se traduce en un **aumento de la productividad**.” (UNIR 2021)
- **ISTQB:** Son las siglas de “International Software Testing Qualification Board” organización internacional sin fines de lucro, creada en noviembre de 2002, en Edimburgo, y registrada legalmente en Bélgica, está dedicada al avance en la profesión de pruebas y se ha convertido en el líder mundial en Certificación de pruebas de software. (2023)

- **Macro:** En Excel, macro es una secuencia de instrucciones that automatiza algún aspecto de Excel tanto que puede trabajar mas eficientemente y con pocos errores. (Alexander M. 2017)
- **Mantis BT:** definida en su página web como “a popular open source web-based bug tracking system. Built on PHP and supports Linux, Windows and macOS on the server side. Compatible with Chrome, Firefox, Safari, Opera and IE 10+. Released under the terms of the GNU General Public License (GPL).” (Mantis BUG TRACKER 2000-2022)
- **POM:** Término cuyas siglas son Page Object Model definido como “un patrón de diseño de objetos en Selenium, donde las páginas web se representan como clases y los diversos elementos de la página se definen como variables en la clase.” (Tutorial Selenium, 2019)
- **Pull Request:** Desde el punto de vista de Git significa que cuando tu trabajo es empujado a tu tenencia, se necesita que se notifique al mantenedor. Como mantenedor llamemos integrador y/o revisor del código. (Chacón S. 2014)
- **REDIS:** Según su propia documentación “Redis is an open source (BSD licensed), in-memory **data structure store** used as a database, cache, message broker, and streaming engine. Redis provides data structures such as strings, hashes, lists, sets, sorted sets” (Redis Ltd. 2022)
- **Rest Assured:** Se describe así mismo como un lenguaje específico de dominio de JAVA para simplificar las pruebas de APIs REST basado en servicios. (Richardson A. 2017)
- **REST:** Arquitectura de APIs donde usualmente responde a HTTP verbos como comandos. (GET, POST, PUT, DELETE):
 - GET: Para obtener información
 - POST Para crear información
 - PUT: Para modificar información
 - Delete: Para eliminar información. (Richardson A. 2017)
- **Retail:** Venta al minoreo.

- **NEW RELIC ONE:** “Una plataforma capaz de escalar masivamente y que recolecta y contextualiza todos los datos operativos sin importar de dónde vengan y simplifica la instrumentación, la ingestión de datos, la exploración, la correlación y el análisis basado en aprendizaje automático (machine learning), para reforzar la observabilidad de cada organización.” (New Relic, Inc 2018-2022)
- **Review:** Ceremonia correspondiente al marco SCRUM donde “Los miembros del equipo principal de Scrum y los stakeholders relevantes participan en las reuniones de revisión del sprint para aceptar los entregables que cumplan con los criterios de aceptación de las historias de usuario y rechazar los entregables no aceptables. Tales reuniones se convocan al final de cada sprint. El Equipo Scrum demuestra los logros del sprint, incluyendo las nuevas funcionalidades o los productos elaborados” (Guía SBOK 2017 p. 258)
- **Screen Mirror:** Aplicación que permite reproducir la pantalla de un dispositivo móvil en una computadora.
- **Scrum Master:** “líder servicial” del Equipo Scrum y es quien modera y facilita las interacciones del equipo como coach y motivador del mismo. Este rol es responsable de asegurarse que el equipo tenga un ambiente de trabajo productivo protegiéndolo de influencias externas, eliminando todos los obstáculos y haciendo que se cumplan los principios, aspectos y procesos de Scrum” (Guía SBOK 2017 p.51)
- **Smoke Test:** Son pruebas que se realizan a un producto software para evaluar su estabilidad y calidad antes de que el software sea sometido a pruebas más exhaustivas y con más detalle. “La idea, es hacer un repaso rápido de lo más crítico que se va a entregar y saber que funciona de manera correcta. No deben de encontrarse defectos críticos o bloqueantes y si no, debería de pararse la build hasta su resolución.” (QALovers, 2021)
- **UAT:** Definido en sus siglas como User Acceptance Testing que en español significa: Pruebas de aceptación de usuario, último filtro de calidad realizado por usuarios actuales, idealmente expertos que conocen más del problema del negocio y resuelto por el sistema (Black R. 2011 p. 473)

REFERENCIAS

Advanced Software Testing Vol. 2. Guide to the ISTQB Advance Certification as Advanced Test Manager (Vol. 2). (2009). Rocky Nook.

Advancing Computing as a Science & Profession. (2018, 22 de Junio). Código de Ética y Conducta Profesional de ACM.
<https://www.acm.org/about-acm/code-of-ethics-in-spanish>.

Alexander, M. (2017). *Excel Macros For Dummies*. John Wiley & Sons.

Avdiji, H., Elikan, D., Missonier, S., & Pigneur, Y. (2020). A design theory for visual inquiry tools. *Journal of the Association for Information Systems*, 21(3), 695–734. <https://doi.org/10.17705/1jais.00617>

Becker, P., Fowler, M., Beck, K., Brant, J., Opdyke, W., & Roberts, D. (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional.

Cardozo, L. (2023, June 29). API: ¿Qué es API, ejemplos y su importancia! *Zenvia*.
<https://www.zenvia.com/es/blog/que-es-api/>

CEPAL & Naciones Unidas. (2022). Tecnologías digitales para un nuevo futuro.
Agenda Digital Para América Latina Y El Caribe.

CodeceptJS. (n.d.). <https://codecept.io/>

Communications. (2017, July 7). “Las oportunidades que abre la digitalización dan más poder al cliente.” *BBVA NOTICIAS*. <https://www.bbva.com/es/oportunidades-abre-digitalizacion-dan-mas-poder-cliente/>

Design Thinking: Developing Innovative Solutions to Make a Difference. (2022). *Delta Kappa Gamma Bulletin*.

Estrella, L., & Schultz, L. E. (2009). *El oráculo del guerrero*.

Gironés, J., Tirado B. (2018). *El Gran Libro de Android*. (7ma Ed.) Marcombo.

International Software Testing Qualifications Board). (2014). *Certified Tester Foundation Level – Agile Tester (CTFL-AT)* <https://astqb.org>

International Software Testing Qualifications Board. (2018). *Certified Tester Foundation Level. (CTFL)* (v3.1.1). <https://astqb.org>

- Introduction to Redis.* (n.d.). Redis. <https://redis.io/docs/about/>
- Lujan J. (2019, February 6). *Page Object Model y Page Factory - Tutorial Selenium.* Tutorial Selenium. <https://www.tutorial selenium.com/2019/02/05/page-object-model-selenium-webdriver/>
- Mantis BT Team. (Copyright 2000-2022). *Mantis Bug Tracker.* <https://www.mantisbt.org/demo.php>
- Morales M., Garzías J. (2016) Gherkin... Una guía de supervivencia ágil. 233 Grados de TI.
- New Relic One. (s.f.). *New Relic One: Una plataforma de observabilidad sencilla pero potente.* <https://newrelic.com/es/resources/datasheets/new-relic-one>
- Osterwalder A., Pigneur Y, Bernarda G.,Smith A. (2020, Marzo). Diseñando la propuesta de valor Ed. Planeta. 1ra Edición.
- QA Lovers. (2021, 25 octubre). *Que es Smoke Test?.* <https://www.qalovers.com/2017/12/smoke-test.html>
- Richardson A. (2017) Automating and Testing a REST API -A Cases Study using Java, REST Assured, Postman, Tracks, CURL and HTTP Proxies
- Ries, E. (2011). *El método Lean Startup: Cómo crear empresas de éxito utilizando la innovación continua.* Deusto.
- Rogers, D. L. (2021). *Guía estratégica para la transformación digital: Reinventa tu negocio para la era digital* (1ra ed.). Empresa Activa.
- Satpathy T. (2017) *Una guía para el CUERPO DE CONOCIMIENTO DE SCRUM* (Guía SBOK™), ScrumStudy, 3ra Edición.
- Soldevila Ll. (Junio 2019) *Digital thinking. Lidera con éxito la transformación digital.* 1ra Ed. Profit.
- Sommerville I. (2002) *Ingeniería de Software.* Pearson Educación.
- Straub, B., & Chacon, S. (2014). *Pro Git.* In *Apress eBooks.* <https://doi.org/10.1007/978-1-4842-0076-6>
- Svalina, A., Tomiša, M., Čačić, M., & Hajdek, K. (2022). Synthesis of current knowledge and research on the design thinking methodology. *Tehnički Glasnik*, 16(4), 445–453. <https://doi.org/10.31803/tg-20210927101638>
- Switek S., Drelichowski L. (2018, 01 June) *Lean Startup – a New Learning Method for Organizations.*
- Unir, V. (2022, May 6). *¿Qué es un IDE en programación? UNIR.* <https://www.unir.net/ingenieria/revista/ide-programacion/>

BIBLIOGRAFÍA

- Black R. (2009) Advanced Software Testing Vol. 2. Guide to the ISTQB Advance Certification as Advanced Test Manager. (Ed. Rocky nook)
- Black R. Mitchell L. (2009) Advanced Software Testing Vol. 3. Guide to the ISTQB Advance Certification as Advanced Test Manager. (Ed. Rocky nook)
- Blanch Torra, L., & Desco Menéndez, M. (2016). *Guía de Buenas Prácticas en gestión de la innovación*.
- Chacón S. & Straub B. (2014) Pro Git, *Everything you need to know about GIT*. Apress.
- CodeceptJS. (2022). *Codeceptjs - Supercharged End 2 End Testing Framework for Nodejs*. <https://codecept.io/>.
- Donaldson, M. (2002). *Inside the Buffer Overflow Attack: Mechanism, Method and Prevention Sans Whitepaper*.
- Fowler M. (1999) Refactoring. Improving the design of Existing Code
- International Software Testing Qualifications Board. (2014). *Certified Tester Foundation Level Agile Extensión (CTFL-AT)*.
<https://www.istqb.org/certifications/agile-tester>
- Carnell J. (2017) Spring Microservices IN ACTION. Manning.
- Mendoza Farro, T. (2017). *Emprendimiento e Innovación*. Universidad Continental
https://repositorio.continental.edu.pe/bitstream/20.500.12394/4260/1/DO_UC_EG_MAI_UC0281_2018.pdf
- Open Web Application Security Project. (2005, 27 julio). *Una Guía para Construir Aplicaciones y Servicios Web Seguros (Ed. 2.0 Black Hat)*.
https://owasp.org/www-pdf-archive/OWASP_Development_Guide_2.0.1_Spanish.pdf
- Rogers D. (Junio 2021) Guía estratégica para la transformación digital. Reinventa tu negocio para la era digital. (1ra edición Empresa Activa)
- Sanchez Cano, G (2018) Seguridad cibernética, Hackeo ético y programación defensiva. (1ra ed. Alfaomega).
- Serrano Ortega, M., & Blazquez Ceballos, P. (2014). *Design Thinking, Lidera el presente. Crea el futuro*. ESIC
- Soldevilla Ll. (2019). Design Thinking – Lidera con éxito la transformación digital. PROFIT.
- Sommerville I. (2002) Ingeniería de Software. Pearson Educación. 6ta Edición.

Stair, R., & Reynolds, G. (2010). *Principios de Sistemas de información. Un enfoque administrative* (9.^a ed.). CENGAGE Learning.



DESARROLLO PROFESIONAL HACIA EL "FUNCIONAL TESTING" Y ANÁLISIS DEL NEGOCIO

INFORME DE ORIGINALIDAD

13%

INDICE DE SIMILITUD

12%

FUENTES DE INTERNET

1%

PUBLICACIONES

7%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

tupropiaescuela.wordpress.com

Fuente de Internet

3%

2

www.coursehero.com

Fuente de Internet

2%

3

Submitted to Universidad de Lima

Trabajo del estudiante

1%

4

www.unizar.es

Fuente de Internet

1%

5

idoc.pub

Fuente de Internet

1%

6

www.acm.org

Fuente de Internet

1%

7

docplayer.es

Fuente de Internet

1%

8

newrelic.com

Fuente de Internet

<1%

9

docs.mantisbt.org

Fuente de Internet