

**PRÁCTICA EN AUTOMATIZACIÓN DE PRUEBAS DE USUARIO PARA EL
GRUPO IES & CMT ACCENTURE COLOMBIA**

INFORME FINAL

SEBASTIAN AUGUSTO BAQUERO PEÑA.

**DANNY HERNANDEZ
ING. SISTEMAS**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
BUCARAMANGA
2018**

**PRÁCTICA EN AUTOMATIZACIÓN DE PRUEBAS DE USUARIO PARA EL
GRUPO IES & CMT ACCENTURE COLOMBIA**

SEBASTIAN AUGUSTO BAQUERO PEÑA.

***PRÁCTICA EMPRESARIAL PRESENTADA PARA OPTAR AL TÍTULO DE
INGENIERO ELECTRÓNICO***

***Docente Supervisor:
MSc. Sergio Alexander Salinas***

***Supervisor Accenture:
Ing. Danny Hernandez***

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
BUCARAMANGA
2018**

*“Dedicado a Dios, mis padres, hermanos,
novia, familia y amigos quienes han sido de gran apoyo
en mi proceso formación y crecimiento personal
para alcanzar mis metas”.*

AGRADECIMIENTOS

Agradezco a Dios primeramente quien permitió que fuera posible realizar la pasantía. También, agradezco a la Universidad Pontificia Bolivariana quien con sus pilares de formación permitieron desarrollar en mi habilidades, destrezas y herramientas para brindar un buen desempeño en la práctica. Seguidamente, agradezco a Accenture quien me dio la oportunidad de realizar la práctica allí, también aquellos compañeros de trabajo que me brindaron una mano y me orientaron. Así mismo, agradezco a mi director del proyecto quien con su calidad humana estuvo presente para consolidar la práctica y quien durante el periodo académico contribuyó en brindarme las herramientas necesarias que fueron de gran relevancia en la práctica.

Agradezco a mis padres y hermanos quienes me han apoyado a lo largo de toda mi vida guiándome, aconsejándome, orientándome siendo de gran ayuda para salir adelante y por ser el motor que me impulsa a lograr mis sueños y metas. También les agradezco porque han estado ahí para guiar mi crecimiento espiritual, social e intelectual.

Agradezco a mi novia quien con su gran amor, ternura, alegría, sinceridad y sencillez ha sido de gran apoyo para mí sobre todo en los momentos más difíciles, siendo mi motor para salir adelante. Su gran sabiduría para dar consejos ha sido de gran valor para mí, agradezco también a mi novia su incondicionalidad para conmigo desde el periodo académico de la universidad hasta el desarrollo de la práctica que ha permitido tener una mayor fortaleza frente a todo.

Agradezco a mis amigos quienes estuvieron pendiente durante todo el proceso de la práctica y me dieron valiosos consejos.

Finalmente, agradezco a los profesores y compañeros de la universidad quienes hicieron parte no solo de mi proceso de formación académicos sino también de mi proceso personal e integral.

CONTENIDO

1. INTRODUCCIÓN.....	13
2. DATOS DE LAS EMPRESAS.....	14
2.1. Accenture Ltda.	14
Acerca de Accenture.	14
2.1.1. Accenture Soluciones.	14
2.1.2. Accenture Industrias	15
2.2. América Móvil	16
2.3. Claro Colombia	16
3. OBJETIVOS.....	17
3.1. Objetivo General.....	17
3.2. Objetivo Específico	17
4. PROYECTO UNIFICACIÓN DE PRECIOS.....	18
4.1. Descripción del Proyecto Unificación de Precios	18
4.2. Operación.....	18
4.3. Definición del problema.....	19
5. MARCO TEÓRICO.....	20
5.1. Sector de Telecomunicaciones en Colombia.....	20
5.2. Entidades que Regulan al Sector de Telecomunicaciones.....	20
5.2.1. Superintendencia de Industria y Comercio.	20
5.2.2. El Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC).....	20
5.2.3. Comisión de regulación de Comunicaciones (CRC)	21
5.2.4. Autoridad Nacional de Televisión (ANTV).....	21
5.2.5. Superservicios	21
5.3. Metodologías	21
5.3.1. Metodologías Ágiles.....	21
5.4. Internet	27
5.4.1. Bases Web.....	27
5.4.2. HTTP.....	28
5.4.3. HTML	28
5.5. Aplicaciones Web.....	29
5.5.1. ¿Cómo trabaja una aplicación Web?	29
5.5.2. Fases del Desarrollo de una Aplicación Web.....	30
5.5.3. Herramientas	31
5.6. Automatización de Pruebas.....	31
5.6.1. Principios para la Automatización	31

5.6.2. Proceso de Automatización de Pruebas	32
5.7. Herramientas.....	35
5.7.1. Visual Studio	35
5.7.2. Visual Studio Team Foundation Services	35
6. MATERIALES Y MÉTODOS.....	36
6.1. Materiales	36
6.2. Metodologías	36
6.2.1. Scrum.....	36
6.2.2. DevOps.....	40
7. IoT.....	43
7.1. Definición	43
7.2. Redes y Comunicación	43
7.3. Gestión de Datos	44
7.4. Aplicaciones	44
7.5. Planteamiento del Proyecto.....	45
7.5.1. Fase I.....	45
7.5.2. Fase II.....	46
7.5.3. Fase III.....	46
7.6. Desarrollo del Proyecto	47
7.6.1. Arquitectura.....	47
7.6.2. Ejecución	48
7.6.3. Producto Final.....	60
8. DESARROLLO DE AUTOMATIZACIÓN DE PRUEBAS.....	61
8.1. Descubrimiento	62
8.2. Planeación.....	62
8.3. Desarrollo.....	63
8.4. Pruebas	66
8.5. Implementación	67
9. RESULTADOS.....	68
10. CONCLUSIONES	71
11. BIBLIOGRAFÍA	72

Lista de Figuras

	<u>Pág.</u>
Figura 1. Logo Accenture - Accenture Ltd.	14
Figura 2. Sector Industrias - Accenture.....	15
Figura 3. Logo América Móvil.	16
Figura 4. Logo Claro.	16
Figura 5. Esquema operación proyecto.	18
Figura 6. Servicios Entidades del Sector de Telecomunicaciones.....	20
Figura 7. Modelo en Cascada.	22
Figura 8. Metodología Ágil.	22
Figura 9. Metodología Scrum.....	25
Figura 10. Metodología Kanban.....	25
Figura 11. Metodología 'Design Thinking'.	26
Figura 12. Metodología 'Lean Start-Up'.	26
Figura 13. Metodología DevOps.	27
Figura 14. Protocolo HTTP.	28
Figura 15. Métodos de Información HTTP.	28
Figura 16. Estructura HTML.....	29
Figura 17. Flujo Básico Aplicación Web.....	30
Figura 18. Fases del Desarrollo de una Aplicación Web.	30
Figura 19. Pruebas Funcionales.	33
Figura 20. Pruebas de Desempeño.	33
Figura 21. Pruebas de regresión.....	34
Figura 22. Automatización de Pruebas.	34
Figura 23. Visual Studio IDE.	35
Figura 24. Visual Studio Team Foundation Services.	35
Figura 25. 'Scrum Team'.	37
Figura 26. 'Scrum Framework'.	38
Figura 27. Herramientas - Parte I.....	42
Figura 28. Herramientas DevOps - Parte II.....	42
Figura 29. IoT Communications.	43
Figura 30. Aplicaciones IoT.	44
Figura 31. Metodología.	45
Figura 32. Módulo NodeMCU ESP8266.	47
Figura 33. Arquitectura Planteada.	48
Figura 34. Inclusión 'package' ESP8266.....	49
Figura 35. Inclusión librerías NodeMCU y Ubidots.	49
Figura 36. Funcionamiento On/Off y Variación Intensidad.....	51
Figura 37. Librería DTH11.	52
Figura 38. Humedad y Temperatura en Ubidots.....	55
Figura 39. Control Servo en Ubidots.....	60
Figura 40. Prototipo terminado.....	60
Figura 41. Prototipo en funcionamiento.	61
Figura 42. Estructura de Trabajo.	61
Figura 43. Flujo básico pruebas.....	62
Figura 44. Estructura componentes automatizaciónn.	63

<i>Figura 45. Arquitectura de automatización de pruebas.....</i>	<i>63</i>
<i>Figura 46. Archivo Excel.</i>	<i>64</i>
<i>Figura 47. Diagrama de clases.</i>	<i>64</i>
<i>Figura 48. Desarrollo - Parte I.....</i>	<i>65</i>
<i>Figura 49. Desarrollo - Parte II.....</i>	<i>65</i>
<i>Figura 50. Desarrollo - Parte III.....</i>	<i>66</i>
<i>Figura 51. Ciclo Automatización de Pruebas.</i>	<i>66</i>
<i>Figura 52. Implementación.</i>	<i>67</i>
<i>Figura 53. Reporte final TFS.....</i>	<i>68</i>
<i>Figura 54. Reporte Local.</i>	<i>68</i>
<i>Figura 55. Reporte formato 'HTML'.....</i>	<i>69</i>
<i>Figura 56. Tiempos de una prueba automatizada.....</i>	<i>69</i>

Lista de Tablas

Pág.

<i>Tabla 1. Comparación de enfoques de Desarrollo de Software.....</i>	<i>23</i>
--	-----------

GLOSARIO

IES: Intelligent Engineering Services.

IoT: Internet of Things.

DevOps: Development and Operations.

VS: Visual Studio.

TFS: Team Foundation Server.

Selenium: Producto para la automatización de pruebas funcionales implementado en Visual Studio.

HTTP: Hypertext Transfer Protocol.

HTML: HyperText Markup Language

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: PRÁCTICA EN AUTOMATIZACIÓN DE PRUEBAS DE USUARIO PARA EL GRUPO IES & CMT ACCENTURE COLOMBIA

AUTOR(ES): SEBASTIÁN AUGUSTO BAQUERO PEÑA

PROGRAMA: Facultad de Ingeniería Electrónica

DIRECTOR(A): SERGIO ALEXANDER SALINAS

RESUMEN

La práctica empresarial en Accenture Colombia consistió en el diseño e implementación de un prototipo de domótica que involucra la tecnología 'Internet de las Cosas' como prueba concepto para presentar los miembros del grupo IES de Accenture con el objetivo de hacer uso de metodologías ágiles. Adicionalmente, en la práctica se apoyó al equipo de desarrollo en la automatización de pruebas funcionales para el proyecto 'Venta de Tecnología y Terminales' en la aplicación web de Claro con el propósito de agilizar la validación del correcto funcionamiento para ser implementado al área de negocio. En este proyecto se hizo el reconocimiento de la plataforma, se recopiló la información necesaria para las pruebas automatizadas y finalmente se hizo el desarrollo.

PALABRAS CLAVE:

Domótica, Aplicación Web, Automatización de pruebas, Telecomunicaciones, Visual Studio.

V° B° DIRECTOR DE TRABAJO DE GRADO

GENERAL SUMMARY OF WORK OF GRADE

TITLE: INTERNSHIP IN USER TEST AUTOMATION FOR IES & CMT
ACCENTURE COLOMBIA

AUTHOR(S): SEBASTIÁN AUGUSTO BAQUERO PEÑA

FACULTY: Faculty of Electronic Engineering

DIRECTOR: SERGIO ALEXANDER SALINAS

ABSTRACT

This internship was done in Accenture Colombia, it consisted to design and develop a home automation prototype that involves Internet of Things Technology like a concept test for presenting to Intelligent Engineering Services' Accenture group with the objective to use agile methodologies. Additionally, in the internship, I supported the development team with the Functional Automation Tests for Sale of Terminals and Technology project onto web application of Claro with the purpose to speed up the functional validation for being implemented in the business area. It was made a platform recognition, gathered information needed for automation tests and we made the development.

KEYWORDS:

Home Automation, Web Application, Test Automation,
Telecommunications, Visual Studio.

V° B° DIRECTOR OF GRADUATE WORK

1. INTRODUCCIÓN

El incremento exponencial de la tecnología, tanto en dispositivos como herramientas, ha generado que las empresas, particularmente del sector de las telecomunicaciones, sientan la necesidad de involucrarse y estar a la vanguardia generando cambios a nivel corporativo y funcional, enfocados siempre en el incremento de su facturación a través de nuevos servicios y productos.

Claro, compañía líder en telecomunicaciones a nivel de Latinoamérica opera en más de 18 países. Particularmente, en Claro Colombia cuentan con más de 30 millones de líneas activas y se ha enfocado no sólo en ofrecer servicios de telefonía móvil, televisión, banda ancha sino también en tecnología ampliando su mercado de oferta. Recientemente se ha incursionado en el campo de 'Internet de las Cosas' y servicios en la nube, siendo uno de los pioneros en el sector de las Telecomunicaciones en Colombia en ofrecer esa tecnología.

A raíz de la expansión de centros de atención de ventas autorizados y venta por telemercado, nace la necesidad de manejar precios fijos estandarizados para todos los puntos de tal manera que no genere aspectos negativos a nivel del cliente y su vez les permita mejorar la gestión interna y contabilización.

Dicho lo anterior, con el uso de automatización de plataformas especialmente de carácter web, se plantea un gran cambio cultural a nivel de la compañía implementando metodologías ágiles, de tal manera que permita generar constante comunicación y participación entre las diferentes áreas generando un mayor rendimiento para ejecución de nuevos productos.

Por ese motivo, se realizó una planeación para el desarrollo e implementación de automatización de pruebas funcionales para una plataforma web para servir de apoyo en los procesos de validación interno de las nuevas pruebas que Claro propone para ofrecer al mercado.

2. DATOS DE LAS EMPRESAS

2.1. Accenture Ltda.

Accenture Ltda, es una empresa multinacional de soluciones la cual está representada por el logo que se muestra en la figura 1.

Figura 1. Logo Accenture - Accenture Ltd.



Tomado del link: https://www.accenture.com/lu-en/careers/jobdetails?id=00593653_en&title=SAP+Cloud+Platform+Integration+Senior+Analyst+-+Midwest+Region.

Acerca de Accenture.

Accenture ofrece a sus clientes solución a grandes retos a través de servicios líderes en Estrategia, Consultoría, Digital, Tecnología y Operaciones. Cuenta con aproximadamente 425.000 empleados con clientes en más de 200 ciudades en 120 países alrededor del mundo [1].

Además, desarrollan e implementan soluciones tecnológicas para mejorar la productividad y eficiencia de sus clientes y ejecutar parte de sus operaciones [1].

2.1.1. Accenture Soluciones.

Accenture cuenta con una gran experiencia en muchos sectores y áreas de negocio ofreciendo grandes resultados de transformación tecnológica a través de las siguientes ramas estratégicas de intervención:

➤ *Accenture 'Strategy'*

Accenture 'Strategy' se encarga de asegurar el éxito de sus clientes mediante la combinación de su gran experiencia en estrategia de negocios, tecnología y operación junto con los modelos de negocios [1].

➤ *Accenture 'Consulting'*

Accenture 'Consulting' ofrece a sus clientes diseños e implementación de soluciones de transformación de negocios y herramientas necesarias para tomar decisiones disruptivas que los ayuden a construir su futuro [1].

➤ *Accenture Digital*

Accenture Digital brinda las mejores experiencias a sus clientes creando nuevos productos y modelos de negocios, mejorando sus capacidades y conexiones con el mundo digital a través de la combinación de sus capacidades en marketing digital, móvil y analítico [1].

➤ *Accenture 'Technology'*

Accenture 'Technology' ayuda a sus clientes a mantenerlos a la vanguardia del mundo digital e impulsarlos a realizar una transformación hacia las nuevas tecnologías [1].

➤ *Accenture 'Operation'*

Accenture 'Operation' apoya a sus clientes a acelerar la innovación y revolucionar los mercados con el uso de servicios en la nube, seguridad, infraestructura, procesos de negocios y capacidad [1].

2.1.2. Accenture Industrias

Accenture abarca varios sectores de la industria en todo el mundo tales como: Automotriz e Industrial, Banca, Mercados Capitales, Química, Energía, Transporte, etc. Permitiendo que la experiencia de los diferentes equipos de trabajo en los diferentes campos impacte positivamente a cada una de ellas ayudándoles a encontrar solución a sus desafíos [1].

Las industrias especializadas a las que Accenture interviene en Colombia son las que se muestra en el diagrama de la figura 2.

Figura 2. Sector Industrias - Accenture.



Basado en la referencia [1].

2.2. América Móvil

América Móvil es una empresa líder de servicios en Latinoamérica y es la cuarta empresa a nivel mundial en número de suscriptores móviles. Ofrece un portafolio de servicios y soluciones en 25 países de América y Europa Central y Europa del Este. Actualmente cuenta con cerca de 362 millones de líneas de acceso, 279 millones de suscriptores móviles, 33 millones de líneas fijas, 28.6 millones de accesos banda ancha y 21.5 millones de unidades de TV pago [2]. Su logo característico se muestra en la figura 3.

Figura 3. Logo América Móvil.



Tomado del link: <https://www.xataka.com.mx/telecomunicaciones/accionistas-de-america-movil-aprueban-la-creacion-de-telesites>

2.3. Claro Colombia

Es una empresa de telecomunicaciones (conocida antes como 'Telmex' y 'Comcel') subsidiaria de un grupo mexicano llamado 'América Móvil'. Claro es un proveedor líder de telecomunicaciones en América Latina con operación en 18 países ofreciendo una gran variedad de servicios de telefonía fija, móvil, televisión, internet, consolas [3]. Claro se identifica con el logo que se observa en la figura 4.

Figura 4. Logo Claro.



Tomado del link: <https://twitter.com/clarocolombia>

3. OBJETIVOS

3.1. Objetivo General

- ✓ Desarrollar e Implementar pruebas de usuario automatizadas en una aplicación Web teniendo en cuenta las metodologías ágiles empleadas por Accenture para visualizar de forma rápida los hallazgos, reducir tiempos en ejecución de pruebas de calidad y lograr una estabilización rápida.

3.2. Objetivo Específico

- ✓ Identificar las principales metodologías Ágiles utilizadas por Accenture.
- ✓ Diseñar e implementar un prototipo de proyecto a baja escala en el campo de las nuevas tecnologías aplicando una de las metodologías ágiles empleadas por Accenture como herramienta base para la automatización de pruebas.
- ✓ Seleccionar la herramienta óptima para el desarrollo de automatización de pruebas.
- ✓ Evaluar el desempeño obtenido durante y después de la automatización de pruebas.

4. PROYECTO UNIFICACIÓN DE PRECIOS

4.1. Descripción del Proyecto Unificación de Precios

El proyecto unificación de precios consiste en ofrecer una única oferta de Terminales y Equipos de tecnología en todos los centros de atención de venta y los de domicilio.

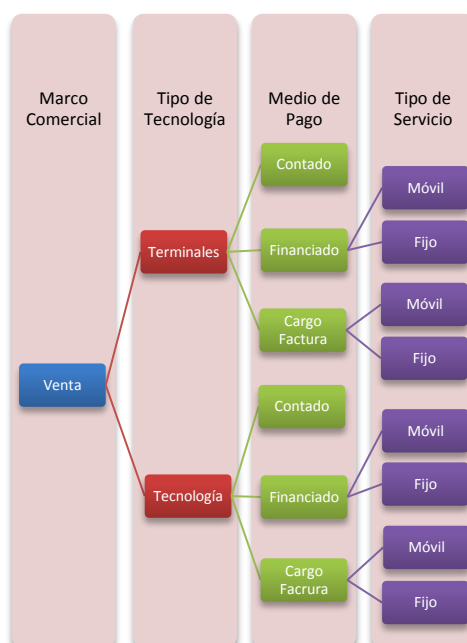
Los objetivos de este proyecto se encaminan en los siguientes aspectos.

- Unificación de las políticas de oferta comercial terminales y tecnología.
- Manejo de una única política de financiación y evaluación comercial.
- Estandarización en los procesos de terminales.
- Unificación de las políticas de oferta comercial y precios para terminales y tecnología.
- Inventario unificado.
- Financiación Terminales y Tecnología.
- Venta puntos directos autorizados.

4.2. Operación

A continuación, se muestra un esquema sobre el funcionamiento del proyecto a nivel global.

Figura 5. Esquema operación proyecto.



Basado en el documento FSP del proyecto de Claro.

4.3. Definición del problema

El proyecto unificación de precios nace como una propuesta a la operación de terminales y tecnologías por parte de Claro Colombia.

Claro Colombia, siendo subsidiaria de América Móvil y líder a nivel Latinoamérica en el negocio de las telecomunicaciones requiere una solución integral para el manejo y operación de los procesos de venta de terminales y tecnologías (operan de forma independiente) a través de un solo sistema.

Por lo anterior, nace el proyecto unificación de precios con el fin de crear un sistema como mecanismo para incorporar todas las dependencias y procesos que hacen parte de la venta de terminales (equipos móviles) y tecnología (televisores, consolas de video juegos, computadores, impresoras, tabletas y demás) de tal manera que facilite su control, manejo y operación. Este sistema tiene como punto de enfoque permitir el manejo de una única oferta comercial en todos los puntos de venta y venta a domicilio de Claro Colombia.

5. MARCO TEÓRICO

5.1. Sector de Telecomunicaciones en Colombia

Colombia en el marco del sector de las tecnologías de la información y la comunicación ha ido aumentando significativamente en los últimos años según el reporte del estudio realizado en el año 2017 según la Comisión de Regulación de Comunicaciones (CRC), destacándose en suscriptores a teléfonos móviles siendo éste el segundo factor más relevante a nivel mundial seguido de los suscriptores activos de banda ancha [4]. Los servicios que ofrecen las entidades del sector de Telecomunicaciones son:

Figura 6. Servicios Entidades del Sector de Telecomunicaciones.



Tomado del reporte 2017 de la Comisión de Regulación de Comunicaciones.

5.2. Entidades que Regulan al Sector de Telecomunicaciones

5.2.1. Superintendencia de Industria y Comercio.

La Superintendencia de Industria y Comercio vela por el buen funcionamiento de los mercados realizando vigilancia y protegiendo la libre competencia económica, los derechos de los consumidores, cumplimiento en el marco legal y reglamentos técnicos. Y también se encarga de proteger los datos personales, vigilancia de las Cámaras de Comercio y asuntos jurisdiccionales [5].

5.2.2. El Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC)

Es una entidad encargada de diseñar, adoptar y promover las políticas, planes, programas y proyectos dentro del marco de las Tecnologías de la Información y las Comunicaciones. Se enfocan en incrementar y facilitar el acceso de todas las personas a las nuevas tecnologías y en sus beneficios promoviendo el uso efectivo y apropiado de ellas [6].

5.2.3. Comisión de regulación de Comunicaciones (CRC)

Es el órgano encargado de promover las competencias, evitar el abuso de posición dominante y regular los mercados de las redes y los servicios de comunicaciones con el propósito de que los servicios se presenten de forma económica y eficiente con altos niveles de calidad [7].

5.2.4. Autoridad Nacional de Televisión (ANTV)

Es una agencia del estado que tiene como objetivo brindar las herramientas para la ejecución de los planes y programas de la prestación del servicio público de televisión con el propósito de velar por su acceso, garantizar el pluralismo informativo, la competencia y la eficiencia del servicio [8].

5.2.5. Superservicios

La Superintendencia de servicios públicos es una entidad técnica que se encarga del mejoramiento de la calidad de vida en Colombia a través de la vigilancia, inspección y control a la prestación de los servicios públicos domiciliarios, la protección de los derechos y la promoción de los deberes de los usuarios y prestadores [9].

5.3. Metodologías

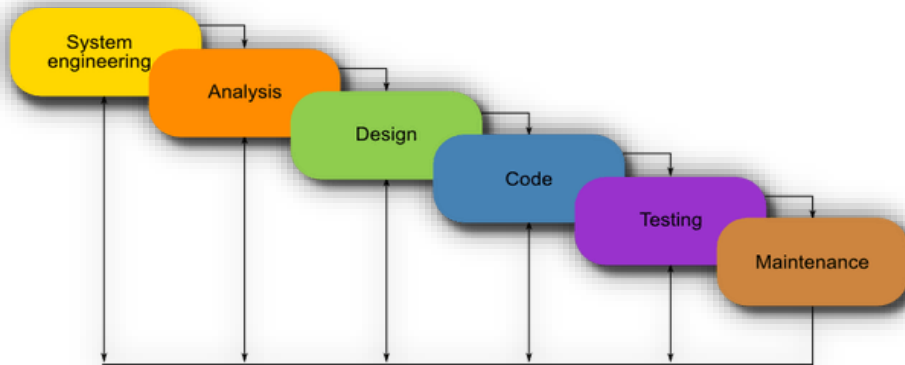
5.3.1. Metodologías Ágiles.

Las metodologías ágiles se enfocan en la gestión de proyectos específicamente en el desarrollo de 'software' empleando iteraciones incrementales llamados 'Sprints' con el fin de facilitar una acción de respuesta rápida frente a un evento imprevisto [10].

Historia

Con el inicio en el campo del desarrollo de software con sistemas de operación se fue convirtiendo rápidamente en un área muy popular en los años 60's y 70's como un nuevo foco de la computación. Comenzando con el proceso de desarrollo de software llamado 'Modelo en Cascada' introducida por SAGE en una conferencia de 'Metodología de desarrollo de Software' el cual consiste en una secuencia de fases de desarrollo en la cuales cada etapa es completada antes de que comience la siguiente etapa con una estructura de foco [11]. En la figura 7 se observa el modelo en cascada con cada una de sus etapas donde se evidencia dependencia en cuanto a que hasta no terminar una de ellas no se puede continuar con la siguiente.

Figura 7. Modelo en Cascada.

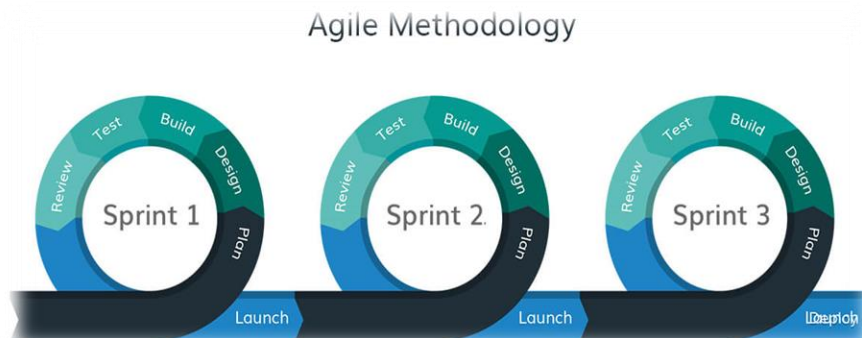


Tomada del link: <https://airbrake.io/blog/sdlc/waterfall-model>.

Más adelante con el incremento de la tecnología y el uso de sitios Web, la sociedad fue creando dependencia con el Internet respecto al manejo de correos electrónicos, e-commerce, y actualizaciones en tiempo real. Por tal razón, equipos de desarrollo de productos se vieron en la necesidad de adoptar un nuevo mecanismo que les permitiera responder rápidamente al cambio del mercado, solución que les llegó en la forma de Desarrollo Ágil con procesos iterativos [11].

Por lo anterior surge la nueva metodología llamada Metodología Ágil la cual se centra en enfoque iterativos para el desarrollo de software, diseñada para permitir flexibilidad y soporte en las necesidades del proyecto y la organización [11]. Vemos en la Figura 8 cómo esta metodología se realiza con iteraciones la cuales busca generar un producto mínimo viable.

Figura 8. Metodología Ágil.



Tomado del link: <https://appinventiv.com/blog/reasons-why-we-trust-agile-for-our-mobile-app-development-process/>

A continuación, se muestra una tabla comparativa entre las Metodologías Tradicionales y las Metodologías Ágiles.

Tabla 1. Comparación de enfoques de Desarrollo de Software.

Enfoques de Desarrollo de Software	
Cascada	Ágil
Prescriptivo	Abstracto
Documentación extensa	Documentación mínima
Secuencial	Continuo
Comunicación Formal	Comunicación Informal
Proceso como foco	Comunicación como foco
Cambios graduales	Cambios rápidos

Los cuatro valores que la comunidad ágil adopta son:

1. **Individuos e interacciones sobre procesos y herramientas:**
Relacionado con generación de vínculos de equipo cercanos, arreglos en el entorno de trabajo y otros procedimientos que impulsen el trabajo en equipo [12]. Por ejemplo: Las personas son más efectivas cuando ellas pueden hablar y trabajar juntas reduciendo así las barreras que pueda existir entre ellos [11].
2. **Software de Trabajo sobre documentación completa:**
Relacionado con lanzamientos nuevos en intervalos frecuentes en desarrollo de código simple, sencillo y avanzado [12]. Por ejemplo: El equipo ágil debería construir versiones de capacidad limitadas de producto donde los clientes y usuarios de negocio pueden probar primero antes de aprobarlo [11].
3. **Colaboración del cliente sobre la negociación del contrato:**
Relacionado con la cooperación y buena relación por parte de los desarrolladores con foco a entregar un valor comercial de forma rápida a medida que inicia el proyecto con el fin de reducir los riesgos de incumplimiento de contrato [12]. Por ejemplo: En el proyecto ágil se debería enfocar en las iteraciones regulares con los clientes con el fin de obtener una retroalimentación y trabajar en conjunto para ajustar la línea de tiempos con los progresos del proyecto [11].
4. **Responder al cambio sobre el plan siguiente:**
Relacionado con los posibles ajustes necesarios durante el ciclo de vida del desarrollo que son considerados por el equipo de desarrollo y los representantes del cliente [12]. Por ejemplo: En un proyecto de desarrollo de software los requerimientos del producto son priorizados al principio de cada iteración a diferencia de hacer todos los requerimientos del producto al principio del proyecto [11].

A continuación, se muestra los principios que se deben tener en cuenta para las prácticas ágiles:

- I. Tener como prioridad hacer que el cliente se sienta satisfecho con entregas continuas de software valioso.

- II. Permitir cambios en los requerimientos aun cuando estos sea tardes como ventajas competitivas del cliente.
- III. Entregar software de trabajo frecuentemente de un par de semanas a un par de meses con preferencia a la escala de tiempo más corta.
- IV. Persona del área de negocio y los desarrolladores deben trabajar diariamente juntos a lo largo del proyecto.
- V. Construir proyectos a través de individuos motivados brindándoles el ambiente y soporte que necesitan y confiar en ellos para obtener el trabajo hecho.
- VI. El más eficiente y efectivo método de transmisión de información para y dentro del equipo de desarrollo es una conversación cara-cara,
- VII. El software de trabajo es la primera medida de progreso.
- VIII. Los procesos ágiles promueven desarrollos sostenibles. Los patrocinadores, desarrolladores y usuarios deben ser capaces de mantener una paz constante indefinidamente.
- IX. La atención continua a la excelencia técnica y buen diseño mejora la agilidad.
- X. La sencillez es esencial.
- XI. La mejor arquitectura, requerimientos y diseños surgen de equipos autoorganizados.
- XII. En intervalos regulares el equipo reflexiona sobre cómo ser más efectivo, luego se sintonizan y ajustan su comportamiento.

Beneficios

Los beneficios de las metodologías ágiles son:

- ❖ Reducción de Costos: Corrección de los errores que se van identificando.
- ❖ Rápida respuesta a cambios: Implementando soluciones sobre la marcha.
- ❖ Equipos de trabajo más involucrados y organizados: Facilita el desarrollo de la creatividad e innovación.
- ❖ Intervención del cliente en el proceso: Activa participación del cliente en cada una de las etapas del proceso aportando ideas y opiniones sobre los resultados que se van entregando progresivamente.
- ❖ Entrega de productos a intervalos: Permite hacer entregas parciales o en bloques con el fin de optimizar recursos y labores de seguimiento y control.
- ❖ Eliminación de tareas innecesarias: Priorizando las tareas de un proceso se identifica las de mayor peso, menor peso y las innecesarias

Los Tipos de Metodologías Ágiles se refiere a las diferentes formas de aplicar los conceptos de agilidad y las más utilizados son:

5.3.1.1 Scrum.

Esta metodología ágil se basa en el desarrollo de un producto incremental donde predomina la planificación completa previo a su ejecución. Su trabajo se realiza

con base a grupos autoorganizados y sus fases de desarrollo se revisan constantemente en lo que se llaman iteraciones. Es la figura 9, se muestra los componentes que conforman una iteración [15].

Figura 9. Metodología Scrum.

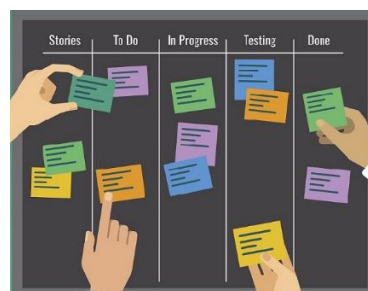


Tomado del link: <http://www.blogandjournal.com/managing-change-with-scrum-methodology-for-web-development/>

5.3.1.2 Kanban.

Es una metodología sencilla y eficaz la cual consiste en medir los tiempos y cantidades en procesos de fabricación para optimizar cada una de las actividades que se llevan a cabo [15]. En la figura 10, observamos un table donde está distribuido por columnas cada una de las etapas para el desarrollo del proyecto en la cual se agregan mediante notas la información correspondiente a cada una ellas.

Figura 10. Metodología Kanban.



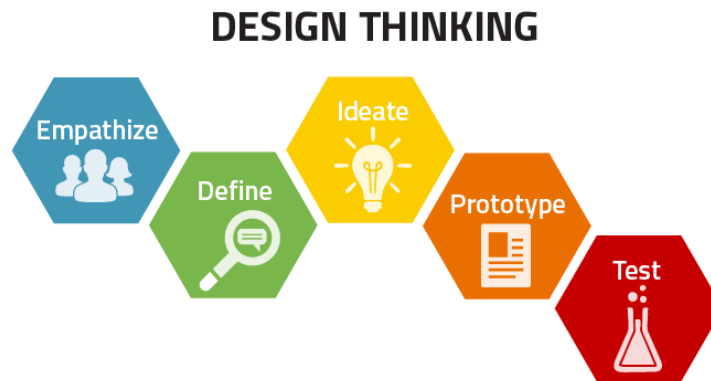
Tomado del link: <http://www.ticbeat.com/empresa-b2b/4-habilidades-para-la-gestion-esenciales-en-la-era-digital-del-futuro/>

5.3.1.3 Design Thinking

Es una metodología ágil basada en el diseño como una herramienta para aportar soluciones al desarrollo de un producto. Busca definir el público a trabajar, comprender sus motivaciones y generar empatía, los cuales son aspectos importantes para esta metodología donde se realiza un proceso creativo del cual se generan todo tipo de ideas previas para luego ser prototipadas y

seguidamente testeadas [15]. En la figura 11, se muestran las etapas de esta metodología empezando por la 'Empatía' hasta la etapa de 'Pruebas'.

Figura 11. Metodología 'Design Thinking'.

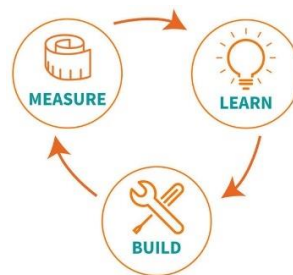


Tomado del link: <https://www.compucom.com/services/design-thinking>.

5.3.1.4 Lean Start-Up

Es una metodología ágil que maneja un proceso de ideación para el lanzamiento de productos basados en iteración o repetición de procesos para obtener un aprendizaje validado con el fin de acortar procesos, adquirir gran conocimiento del mercado que se desea intervenir y reducir los costos iniciales que están asociados al comenzar un negocio [15]. Este proceso lo podemos observar en la figura 12, en la cual está la etapa de aprendizaje, construcción y medición.

Figura 12. Metodología 'Lean Start-Up'.



Tomado del link: https://jtrsolutions.com/blog/customsoftware_leanstartup/.

5.3.1.5 DevOps

Es una metodología ágil formada por un acrónimo de 'Desarrollo y Operaciones' que en inglés es 'Development and Operations' basada en la colaboración,

comunicación e integración especialmente en el desarrollo de software no sin contemplar la posibilidad en aplicarlo a otros modelos de negocio donde se busca generar un entorno o ambiente en el que el lanzamiento de ideas es constante [15]. En la figura 13, se observa cada una de las fases que hacen parte a cada equipo de trabajo tanto para los de 'Desarrollo' como para los de 'Operaciones'.

Figura 13. Metodología DevOps.



Tomado del link: <https://expertise.jetruby.com/the-history-of-devops-and-why-you-should-consider-adopting-it-babeaa778288?gi=a18a07327512>.

5.4. Internet

En 1989 bajo una investigación, Tim Bernes-Lee diseñó un sistema hipertexto para una red de enlaces entre documentos. Posteriormente, consiguió aprobación para continuar con ese proyecto el cual dio origen al primer navegador llamado WoldWideWeb [16].

En 1992 se expande este sistema aumentando la cantidad de servidores estables. Sigue el crecimiento en el año siguiente lanzándose otro navegador 'X-Window/Unix' más adelante llamado 'Netscape'. En 1994 el 'Consortio WWW' se estableció como el estándar en la web, dando lugar al crecimiento continuo de los servicios que conocemos el día de hoy [16].

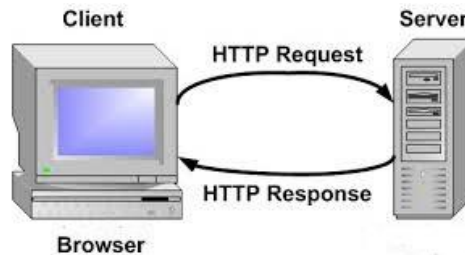
5.4.1. Bases Web

El éxito del internet se debe a dos características básicas: el protocolo 'HTTP' y el lenguaje 'HTML'. La primera permite implementar un sistema de comunicaciones sencillo y directo para que sea fácil enviar cualquier tipo de archivo, haciendo simple el funcionamiento del servidor para permitir que este puede recibir más solicitudes y la segunda es un complemento del protocolo permitiendo dar forma y estructura a las páginas siendo altamente eficiente y fácil de usar [16].

5.4.2. HTTP

Es un protocolo de transferencia de hipertexto que requiere un protocolo de comunicaciones llamado TCP (Transmission Control Protocol) que establece un canal de comunicación de extremo a extremos, es decir, cliente-servidor donde pasan los 'bytes' que constituyen los datos a transferir [16]. En la figura 14 se muestra el intercambio de información.

Figura 14. Protocolo HTTP.



Tomada del link: <https://zju.date/session-vs-cookie/>

Existe una variante de este protocolo y es el protocolo HTTPS, el cual usa un protocolo de seguridad SSL (Secure Socket Layer) para cifrar y autenticar el tráfico entre cliente-servidor, usado con frecuencia para el comercio electrónico, información personal o confidencial [16].

- Métodos de información HTTP: En la figura 15 muestran los métodos más utilizados para el protocolo de comunicación a internet.

Figura 15. Métodos de Información HTTP.

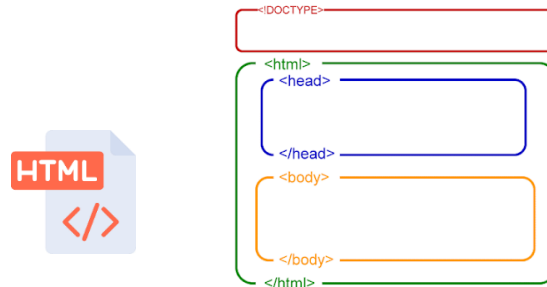
GET	<ul style="list-style-type: none">• Solicitud de recurso para obtener datos del servidor.
POST	<ul style="list-style-type: none">• Solicitud para enviar datos al servidor.
HEAD	<ul style="list-style-type: none">• Solicita los metadatos de un archivo.
PUT	<ul style="list-style-type: none">• Crear/ carga un nuevo recurso o reemplaza uno ya existente.
DELETE	<ul style="list-style-type: none">• Eliminar datos.
TRACE	<ul style="list-style-type: none"><input type="checkbox"/> Ruta de retorno de un mensaje a lo largo de un destino.
OPTIONS	<ul style="list-style-type: none"><input type="checkbox"/> Opciones de comunicación con el servidor.
CONNECT	<ul style="list-style-type: none"><input type="checkbox"/> Establece un túnel hacia el servidor.
•PATCH	<ul style="list-style-type: none">• Modificación parcial

5.4.3. HTML

Es un lenguaje de hipertextos que permite representar el contenido y hacer referencias a otros tipos de recursos tales como documentos, imágenes, videos,

gif's, etc. Además, permite estructurar la aplicación web que se piensa desarrollar de una forma más práctica, sencilla y organizada tal como se apreciará en la siguiente imagen [16]. En la figura 16 se muestra la estructura que contiene una página web en lenguaje HTML.

Figura 16. Estructura HTML.



Tomada del link: <http://dyzain.blogspot.com/2016/03/the-html-template.html>

5.5. Aplicaciones Web

Una aplicación web es cualquier programa de software que una computadora utiliza para una función determinada utilizando un navegador web como cliente o una página web. Esta puede ir de lo simple como un tablero de mensajes o formulario hasta tan compleja como una aplicación de juego multijugador. Además, permite la interacción del usuario y la máquina [17] [18].

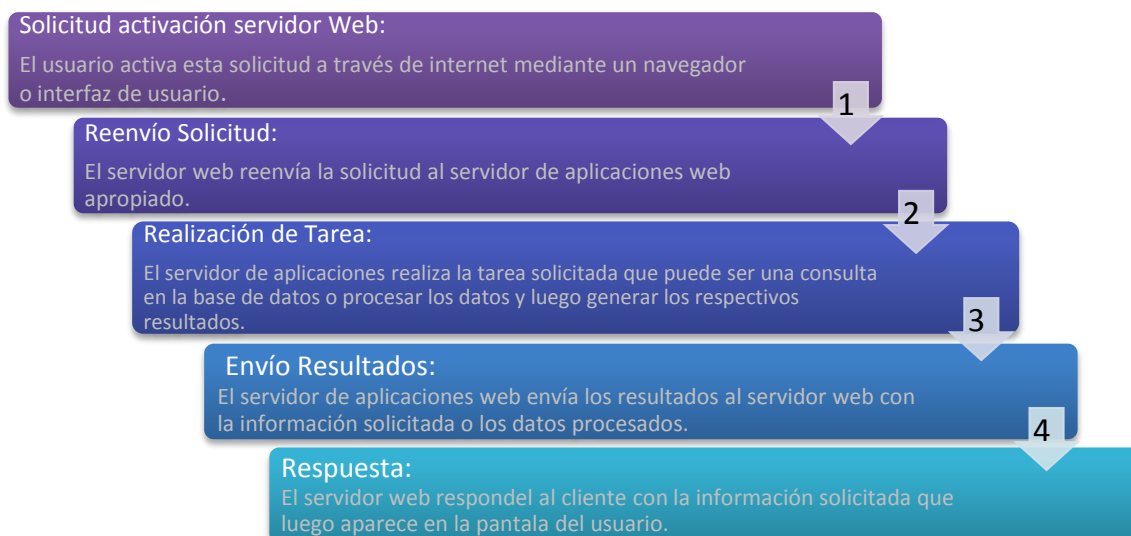
Las aplicaciones web por lo general se crean como producto de una combinación de lenguajes de programación diseñados para uso en internet, y con esto poder reducir el uso de recursos de la memoria RAM para ejecutarse [18]. También una aplicación web es un programa de computador que utiliza buscadores y tecnología web para realizar tareas en internet [19].

5.5.1. ¿Cómo trabaja una aplicación Web?

Una aplicación web es codificada o desarrollada en un lenguaje soportado por el navegador tal como es 'Javascript' y 'HTML'. Algunas aplicaciones son dinámicas y requieren un procesamiento de la información por parte del servidor a diferencia de las estáticas que no requieren de este procesamiento [19].

Las aplicaciones requieren generalmente de tres cosas: servidor para gestionar las solicitudes del cliente, servidor de aplicaciones y una base de datos para almacenar la información [19]

Figura 17. Flujo Básico Aplicación Web.



Tomado de la referencia [19].

5.5.2. Fases del Desarrollo de una Aplicación Web

A continuación, se menciona las diferentes etapas que posee el desarrollo de una aplicación web.

Figura 18. Fases del Desarrollo de una Aplicación Web.



Tomada de la referencia [20].

5.5.3. Herramientas

Las herramientas más comunes y utilizadas son:

- ❖ Ionic: Marco para el desarrollo de aplicaciones móviles y web multiplataforma otorgando: alto rendimiento, implementaciones en vivo, informe de fallas, notificaciones integradas, etc [21].
- ❖ Angular: Plataforma que facilita la creación de aplicaciones con la web que se ejecutan tanto en ella como en dispositivos móviles o escritorio. Además, permite la combinación de plantillas, front-end, back-end, etc [21].
- ❖ TypeScript:
Lenguaje de código abierto basado en programación orientado a objetos, permite agregar funciones como módulos y clases, soportando en muchos editores de código (IDE) [21].
- ❖ Visual Studio Code:
Editor de código fuente y libre para cualquier sistema operativo de PC's, desarrollado por la empresa Microsoft y uso de cualquier lenguaje de programación, herramientas de depuración, fuerte integración y control de Git [21].

5.6. Automatización de Pruebas

Se emplea en muchos campos, uno de ellos es en la automatización de software (aplicación) específicamente de pruebas. Esta automatización se emplea con el fin de reducir las pruebas manuales mejorando la precisión, aumentando la eficacia y eficiencia, cobertura y ahorro de tiempo de las personas encargadas por parte de un equipo de soporte de calidad del software lo que ayuda en la reducción de costos para la compañía [22].

En especial la automatización de pruebas son las más adecuadas en un ambiente en el cual los requerimientos del software varían con frecuencia y se hace necesario la realización de una gran cantidad de pruebas de regresión que si se realizaran manualmente repercute en tardar más tiempo en ejecutarlas [22].

5.6.1. Principios para la Automatización

Las pruebas automatizadas se realizan en diferentes niveles: pruebas unitarias, difusas y funcionales. Estas pruebas tienen principios tal como:

- ❖ *Diseño de pruebas:*

Se crea una clase base encargada de las funciones principales de la aplicación. Con base a las especificaciones se identifica la funcionalidad principal de la

aplicación que cubrirán las pruebas añadiéndole posteriormente más pruebas en paralelo con la implementación [8].

❖ *Identificación de automatización:*

Se debe realizar primero una prueba funcional que sea principal de la aplicación para analizar si es necesario la implementación de pruebas automatizadas. En caso de ser tareas tediosas, repetitivas y de gran manejo de datos si es necesario su implementación [8].

❖ *Pruebas cortas:*

Permiten localizar fácilmente cualquier error o falla presentada en la automatización [8].

❖ *Pruebas independientes:*

Permite organizar las pruebas con clases y sus respectivos métodos de forma separada para no depender de otras pruebas y se pueda ejecutar de forma independiente en cualquier momento [8].

❖ *Legibilidad:*

Encargado de hacer el desarrollo lo más entendible posible para los interesados para ello se recomienda que el código fuente sea autodocumentado [8].

❖ *Pruebas rápidas:*

La velocidad de ejecución de las pruebas debe ser un indicador principal de manejo para así darle mejor calidad a la automatización del software [8].

❖ *Pruebas resistentes al cambio:*

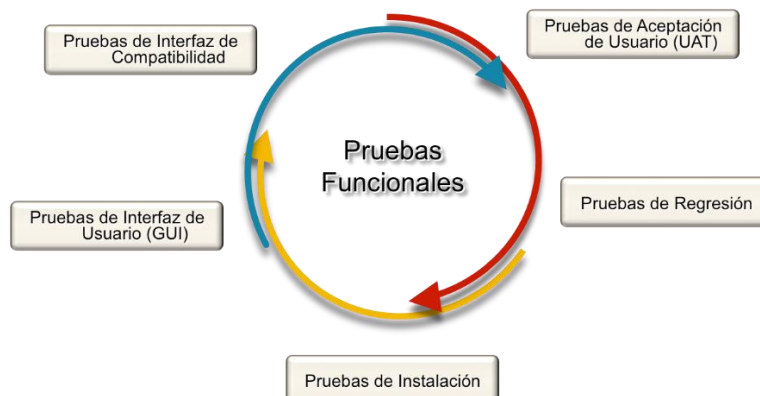
Para evitar los inconvenientes que se presentan al desarrollar pruebas automatizadas que no sean resistentes al cambio en alguna funcionalidad y característica del software se recomienda realizar pruebas de carácter funcional [8].

5.6.2. Proceso de Automatización de Pruebas

5.6.2.1 Pruebas Funcionales

Pruebas que consisten en la disminución del ciclo de prueba y reducción del esfuerzo invertido en ellas proporcionando un ambiente controlado y utilizando herramientas de administración de defectos, de tal manera que reflejen transparencia en los resultados de las pruebas. Estas pruebas de software se usan típicamente para encontrar el comportamiento del sistema, subsistema o componente software especificados en requisitos o casos de uso que pueden o no estar documentados [24] [25]. En la figura 20 se muestra las pruebas que hace parte de las pruebas funcionales.

Figura 19. Pruebas Funcionales.

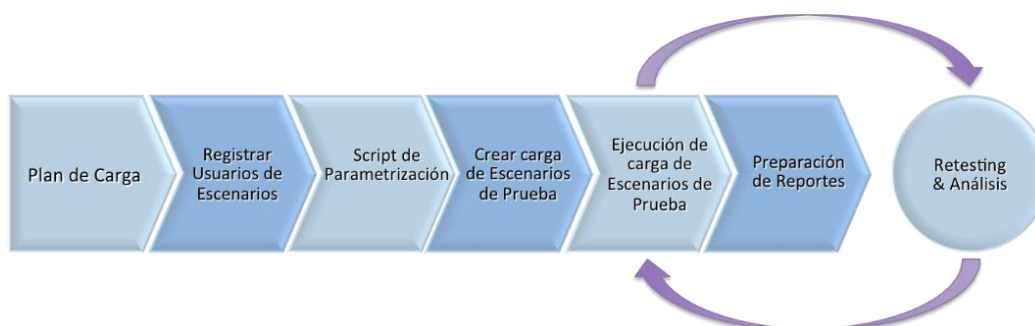


Tomado de la referencia [9].

5.6.2.2 Pruebas de Desempeño

Pruebas en las cuales están inmersas las pruebas de rendimiento, carga, estrés, usabilidad, mantenimiento, fiabilidad o portabilidad que se realizan repetidamente para identificar los límites potenciales del sistema reduciendo las fallas que se puedan presentar en las aplicaciones [24] [25]. Ver figura 20.

Figura 20. Pruebas de Desempeño.

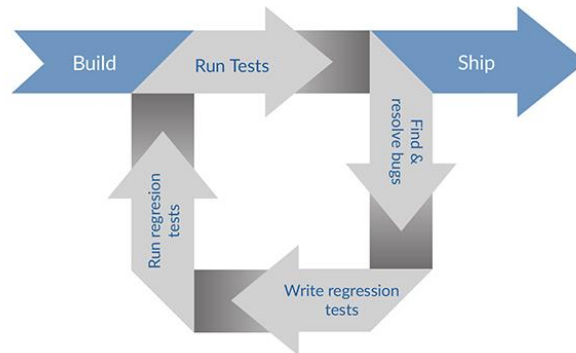


Tomado de la referencia [9].

5.6.2.3 Pruebas de Regresión

Pruebas que consisten en verificar que la funcionalidad básica y el desempeño de la aplicación no se vea afectada por los nuevos cambios realizados, es decir, volver a probar un componente tras haber sufrido una modificación [24] [25]. En la figura 21 se muestra el flujo de las pruebas de regresión.

Figura 21. Pruebas de regresión.



Tomada del link: <https://dzone.com/articles/what-is-regression-testing>

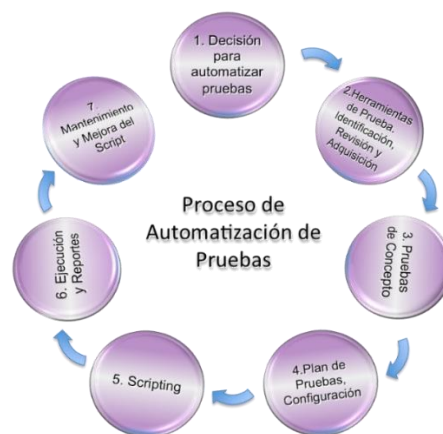
5.6.2.4 Pruebas de Integración

El enfoque de estas pruebas se realiza para verificar que todos los componentes que interactúan en el software estén funcionando correctamente una vez se haya validado cada componente de manera individual [24].

5.6.2.5 Automatización de Pruebas

Estas pruebas se realizan con el propósito de reducir el tiempo por cada iteración del ciclo de la prueba mediante el ahorro en recursos, un ahorrar costos a razón de reducir esfuerzos manuales en las pruebas, mejorar la cobertura en las pruebas de regresión. En estas pruebas pueden estar inmersas algunas o todas las pruebas anteriormente descritas [24]. En la figura 22 se muestra los diferentes pasos para hacer el proceso de automatización de pruebas.

Figura 22. Automatización de Pruebas.



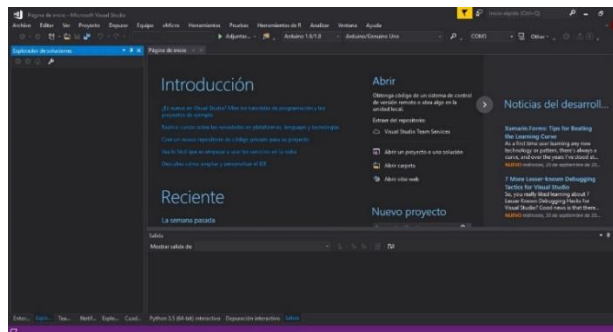
Tomado de la referencia [24].

5.7. Herramientas

5.7.1. Visual Studio

Visual Studio es un Entorno de Desarrollo Integrado (IDE) en el cual es comúnmente utilizado para hacer tareas de desarrollo y pruebas. Además, estos proporcionan un entorno para editar, refactorizar y compilar código, adicionalmente permite editar interfaces de usuario, lógica de diagrama, construir aplicaciones y realizar análisis de código, pruebas e integración de control de código fuente [26]. En la figura 23 se muestra el entorno de desarrollo de Visual Studio versión Community.

Figura 23. Visual Studio IDE.

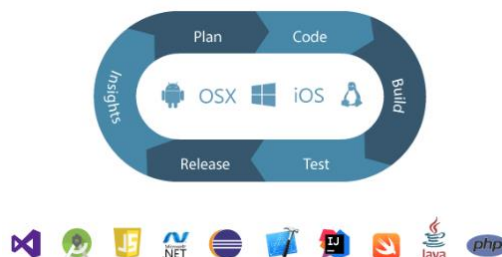


Tomado del link: <https://www.onw4rds.com/visual-studio/>

5.7.2. Visual Studio Team Foundation Services

Es un componente de Visual Studio que proporciona la funcionalidad de colaboración central para los equipos de desarrollo de software en un producto integrado. Esta herramienta permite: Gestión del proyecto, seguimiento del artículo de trabajo (WIT), Control de Versión, Gestión de Casos de Prueba, Automatización de compilación, Informes, Gestión de lanzamiento, Gestión de laboratorio y ambiente, Gestión de comentarios y herramientas de comunicación de chat y equipo [27]. En la figura 24 se muestra las operaciones que realiza Visual Studio Team Foundation Services.

Figura 24. Visual Studio Team Foundation Services.



Tomado del link: <https://visualstudio.microsoft.com/tfs/?rr=https%3A%2F%2Fwww.google.com.co%2F>

6. MATERIALES Y MÉTODOS

6.1. Materiales

Los recursos disponibles en la práctica empresarial, se refiere principalmente a la dotación que facilitó la empresa para el desarrollo de la práctica. Entre los cuales están:

- ✓ Material de información como documentos en Word, PowerPoint, Tablas en Excel e imágenes correspondientes al proyecto.
- ✓ Asignación de un ID de la empresa, mediante el cual se obtuvo acceso a la red global de Accenture y a toda su base de datos de información.
- ✓ Acceso al correo electrónico de la empresa, mediante el cual se accedió a los portales de la empresa y se adquirió información acerca de la metodología o las metodologías utilizadas en los proyectos efectuados por Accenture.
- ✓ Acceso a 'Skype for Business', en el cual se realizaron conferencias y capacitaciones con personal de Accenture.
- ✓ Puesto de trabajo en el edificio de Plaza Claro. Dg. 24c #68-1, Torre 1 Piso 6.
- ✓ Computador portátil por parte de Accenture mediante el cual se realizó las actividades mencionadas anteriormente.

6.2. Metodologías

6.2.1. Scrum

Es una metodología ágil iterativa e incremental basa en equipos para abordar proyectos con el fin de desarrollar cualquier producto o administrar cualquier trabajo. Scrum está constituida por un cierto número de tareas con la participación de un equipo de trabajo que actúa de una manera autoritaria y coordinada para garantizar el cumplimiento del objetivo común y el compromiso con el mismo [29].

Los equipos Scrum de desarrollo se componen por tres roles, lo cuales son:

- Product Owner: Es el punto central de liderazgo con la autoridad de decidir qué funciones y funcionalidades se construirán y de qué orden se llevarán a cabo. Se mantiene en constante comunicación con el equipo para darles una visión más clara de lo que se quiere, por esto tiene la gran responsabilidad de asegurarse que siempre se realice el trabajo más valioso posible respondiendo las inquietudes del Scrum Master y el equipo de desarrollo [30].
- Scrum Master: Ayuda a su equipo de trabajo a aceptar y seguir los valores, principio y prácticas Scrum. Además, siendo líder servicial ayuda a su equipo de trabajo en desarrollar un enfoque de alto rendimiento,

resolver problemas y mejoras eliminando los impedimentos que no permiten que el equipo de trabajo sea productivo [30].

- Development Team: Conformado por un equipo diverso y multifuncional tales como: arquitecto, programador, probador, administrador de bases de datos y diseñador de interfaz de usuario con la responsabilidad de diseñar, construir y probar el producto deseado autoorganizándose para así lograr de una mejor manera cumplir con el objetivo establecido por el 'Product Owner' [30].

En la figura 25 se muestra una representación del equipo de trabajo 'Scrum'.

Figura 25. 'Scrum Team'.



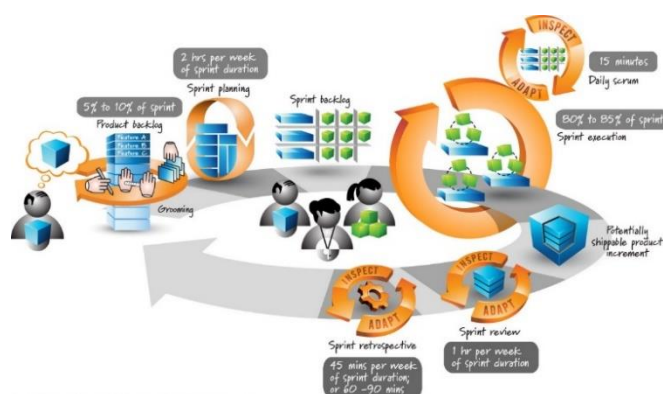
Tomada del link: <https://itnext.io/scrum-and-outsistemas-the-perfect-combination-to-bring-value-do-the-business-5a316f2d34a>.

6.2.1.1 Framework Scrum

El framework Scrum hace referencia a la definición de los roles específicos a utilizar en el marco Scrum.

En ciclo de trabajo Scrum se desglosan las características organizadas en orden de prioridad quien las define él mismo en compañía del Product Owner. Estas características se subdividen en otro grupo de características a realizar por parte del equipo de desarrollo en cada iteración de tal manera que se facilite su desarrollo y ejecución con una planeación previa. Después de haber organizado en subcategorías el equipo de trabajo asigna las tareas de diseño, construcción, integración y prueba de acuerdo con la planeación del respectivo 'Sprint' y realizan reuniones diarias con el fin de hacer una inspección del proceso y hacer ajustes en caso de ser requeridos. Al final cuando se tiene un producto, éste pasa por una revisión y una retrospectiva para analizar si algunos de los pasos ejecutados se pueden replicar en la siguiente iteración. Y así sucesivamente se irán realizando los 'Sprint' de acuerdo con cada subcategoría hasta lograr el producto deseado por el 'Product Owner' [8]. El proceso anterior se puede entender más con la figura 26, que muestra el ciclo completo de trabajo partiendo de los requerimientos de la iteración hasta su retrospectiva o retroalimentación.

Figura 26. 'Scrum Framework'.



Tomada del link: <https://www.innolution.com/blog/how-much-time-should-each-scrum-practice-take>

6.2.1.1.1 Product Backlog

Se conoce también como banco cartera o registro acumulado de producto el cual posee las características necesarias para llevar a cabo el producto deseado. Estas características son organizadas por el 'Product Owner' con la participación del Scrum Master en orden de importancia y prioridad tal como se observa en la Imagen 10, quien a lo largo de las iteraciones puede ir agregando más funciones, hacer cambios en las ya existentes, defectos que necesiten reparación, mejoras técnicas y demás, tal como se evidencia en la Imagen 56. Con lo anterior, se busca velar porque los elementos o características de alto valor aparezcan en la parte superior del 'Product Backlog'. Los elementos pueden ser agregados, eliminados, revisados y modificados de acuerdo con las condiciones del negocio a lo que se le conoce como 'Crecimiento' [30].

Para determinar la prioridad de cada elemento se define el tamaño de cada uno de acuerdo con su costo, el cual no tiene un valor estándar sino es estimado por el equipo vista en la siguiente imagen [30].

6.2.1.1.2 Sprint

Los 'Sprints' también son llamados iteraciones o ciclos con una duración de hasta un mes calendario donde al final de cada uno, se debe crear un producto tangible de valor para el cliente. Estos poseen un intervalo de inicio y uno de finalización de carácter fijo, que una vez se culmine uno de ellos se da paso a la continuación del siguiente ciclo [30]. En la imagen 59, se contrasta cómo se compone cada uno de los 'sprint' y su secuencia.

6.2.1.1.3 Sprint Planning

La planificación de cada iteración o ciclo se realiza en conjunto entre el 'Product Owner', Scrum Master y el 'Development Team' con el fin de organizar las iteraciones de tal manera que no queden sobrecargadas y así poderlas cumplir a cabalidad. Una vez planificado los objetivos del ciclo siguiente, el equipo de

desarrollo revisa la acumulación de productos y determinan los elementos de alta prioridad que pueden lograr cumplir realizando una estimación del esfuerzo que requieren para completar cada una de las tareas obtenidas de cada característica generalmente dada en horas [30].

6.2.1.1.4 Sprint Execution

Comprende el proceso en el cual el equipo de desarrollo guiado por el Scrum Master realiza todo el trabajo necesario para completar las funciones de tal manera que exista un alto grado de confianza para producir características de buena calidad [30].

6.2.1.1.5 Daily Scrum

Scrum diario, se realiza en un tiempo no mayor a 15 minutos donde priman tres preguntas en beneficio de los miembros del equipo las cuales son:

- 1) ¿Qué logré desde el último 'Daily Scrum'?
- 2) ¿En qué planeo trabajar en el próximo 'Daily Scrum'?
- 3) ¿Cuáles son los obstáculos que me impiden hacer progreso?

Estas con el fin de servir de herramienta para comunicar el estado de los elementos del 'Sprint' haciendo una inspección, sincronización y actividad de planificación diaria adaptativa en pro de autoorganizar el equipo para mejor su trabajo [30].

6.2.1.1.6 Sprint Results

Los resultados en cada 'Sprint' son un incremento del producto potencialmente realizable, permite en esta fase especificar el grado de confianza de que el trabajo realizado es de buena calidad y es potencialmente enviable dependiendo de la decisión tomada por la empresa [30].

6.2.1.1.7 Sprint Review

El objetivo de esta actividad es hacer una revisión y adaptación del producto que se está construyendo mediante la comunicación entre los miembros del equipo de trabajo con el fin de tener una clara visibilidad de lo que se hizo para guiar de forma exitosa y apropiada el siguiente desarrollo para el negocio [30].

6.2.1.1.8 Sprint Retrospective

Proceso o etapa en el cual el equipo de trabajo se reúne para analizar lo que funciona y lo que no funciona con miras a buscar un foco y compromiso de mejora continua del proceso [30].

6.2.1.2 Beneficios de las Metodología Ágil Scrum

De acuerdo con ejecutivos de área de Tecnología de 'Forbes Technology Council' mencionan acerca de cómo mediante el uso de metodologías ágiles - 'Scrum' se puede ayudar al equipo a alcanzar sus metas con los siguientes beneficios:

- Ciclo de rápida retroalimentación.
- Constante cambio.
- Identificación temprana de problemas.
- Priorización flexible.
- Alto potencial de satisfacción para el cliente.
- Visualización rápida de beneficios.
- Medición de compromisos y responsabilidad.
- Optimización del tiempo.
- Propósito de trabajo.

6.2.2. DevOps

Es una metodología que aplica los principios ágiles generalmente en todos los proyectos de desarrollo de software, permitiendo que una empresa maximice la velocidad de entrega de un producto o servicio comenzando por la idea hasta el lanzamiento a producción y permite recibir los comentarios de los clientes y hacer mejoras como resultado de esa retroalimentación [32]. En la imagen 67, están ejemplificados los dos equipos que participan en esta metodología, el equipo de desarrollo y el equipo operativo.

Además, la metodología DevOps mejora la forma en cómo la empresa ofrece valor a sus clientes, proveedores y socios. Proporciona un importante retorno de la inversión en áreas cómo:

- Experiencia mejorada para el cliente: Se obtiene producto de responder a los comentarios de los clientes mediante mecanismos de respuesta rápida [32].
- Mayor capacidad para innovar: Permite reducir tiempos y generar mayor valor [32].
- Tiempo más rápido para valorar: Implica el desarrollo de una cultura, prácticas y automatización que permitan una entrega de software rápida, eficiente y confiable hasta producción. También proporciona herramientas necesarias para facilitar la planificación, previsibilidad y el éxito del lanzamiento [32].

6.2.2.1 Principios DevOps

- *Desarrollo y Testeo de sistemas similares a producción:*

Principio cuyo objetivo es permitir que los equipos de desarrollo y aseguramiento de calidad (QA) desarrollen y prueben sistemas que se comporten como el

sistema de producción para tener idea del comportamiento de la aplicación para cerciorarse que su funcionamiento sea el correcto antes de ser implementada [32].

- *Implementación de procesos confiables y repetibles:*

Principio que permite que el desarrollo y las operaciones respalden un proceso de desarrollo ágil hasta la producción. La automatización es importante para crear procesos iterativos, repetibles y confiables la cual debe estar encargada de realizar la organización para ejecutar su implementación y pruebas automáticas permitiendo así que los equipos prueben procesos por sí mismos reduciendo el riesgo de errores en el despliegue [32].

- *Monitoreo y validación de calidad operacional:*

Principio encargado del monitoreo temprano en el ciclo de vida con pruebas automáticas previas proporcionando una alerta temprana sobre los problemas operacionales y de calidad que puedan ocurrir en producción. Y, además, capturando las métricas en un formato entendible y fácil de usar para los interesados [32].

- *Ampliación de bucles de retroalimentación:*

Principio que exige que las organizaciones creen canales de comunicación que permitan a todos los miembros del equipo de trabajo acceder y actuar en la retroalimentación, pero para ello se requiere que ésta se otorgue y actúe de forma rápida [32].

6.2.2.2 Arquitectura

Una arquitectura de referencias es aquella que proporciona una plantilla de una solución mediante el uso de varios métodos y capacidades para diseñar una plataforma que se pueda adaptar al proyecto, para ello existe cuatro rutas [32].

6.2.2.3 Ciclo de vida DevOps

Desarrollo

Proceso de desarrollo subdivido en ciclos como beneficio para acelerar el desarrollo de software y el proceso de entrega [33].

Prueba

El equipo de Asistencia de Calidad (QA) usa herramientas como 'Selenium' para identificar y corregir errores en la nueva pieza de código [33].

Integración

Las nuevas funcionalidades que se van desarrollando se van integrando con el código y pruebas vigentes. Por medio de esta etapa se pueden hacer desarrollos y pruebas continuas [33].

Despliegue

En esta fase el proceso de implementación se lleva a cabo de forma continua de tal manera que cualquier cambio realizado en cualquier parte del código, no se debe ver afectado el funcionamiento en producción [33].

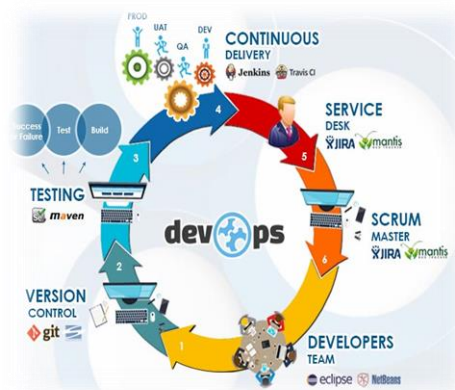
Monitoreo

Permite que el equipo de operación se encargue del comportamiento adecuado del sistema y realice reportes de los errores encontrados en producción [33].

6.2.2.4 Herramientas DevOps

En las figuras 27 y 28 se muestran algunas de las herramientas utilizadas en DevOps.

Figura 27. Herramientas - Parte I.



Tomada del link: <https://habiletechnologies.com/services/devops/>

Figura 28. Herramientas DevOps - Parte II.



Tomada del link: <https://habiletechnologies.com/services/devops/>

7. IoT

7.1. Definición

Internet de las cosas se considera una presencia generalizada en un ambiente de gran variedad de cosas y objetos que a través de conexiones inalámbricas y por cable que con esquemas de dirección única están disponibles para interactuar con cada uno y cooperar con otras cosas y objetos para crear nuevas aplicaciones y servicios para alcanzar metas en común [34].

La meta de esta nueva tecnología es crear ambientes inteligentes que hagan de la energía, el transporte, las ciudades y muchas otras áreas más inteligentes [34].

7.2. Redes y Comunicación

Debido a la cantidad de dispositivos que se conectan a Internet, se hace necesario la creación de una nueva red de Telecomunicación de quinta generación (5G), la cual representa claramente una convergencia de topologías de acceso de red que cuenta con una integración de todas las aplicaciones del IoT dentro de las cuales está los dispositivos electrónicos usables en el cuerpo, sistemas de transporte, plantas de producción, etc [34].

En la figura 29, se observa cómo esta tecnología permite el uso de diferentes datos a través de diferentes mecanismos de transmisión de señales para el manejo de la información.

Figura 29. IoT Communications.



Tomada del link: <http://wiznetmuseum.com/2-simple-iot-gateway/>.

7.3. Gestión de Datos

La gestión de los datos es un aspecto crucial para el 'IoT' debido a que al estar interconectados varios objetos, estos intercambian todo tipo de información constantemente generando un gran volumen de datos y de procesos haciendo difícil su manipulación por tal razón surgen oportunidades para la gestión de esa información dentro de las cuales están:

- ✓ Análisis y Colección de Datos
- ✓ Big Data
- ✓ Red de Sensores Semánticos
- ✓ Sensores Virtuales
- ✓ Procesamiento de Eventos Complejos

7.4. Aplicaciones

En la figura 30 se muestran algunas de las múltiples aplicaciones que se pueden realizar mediante el uso de esta tecnología.

Figura 30. Aplicaciones IoT.



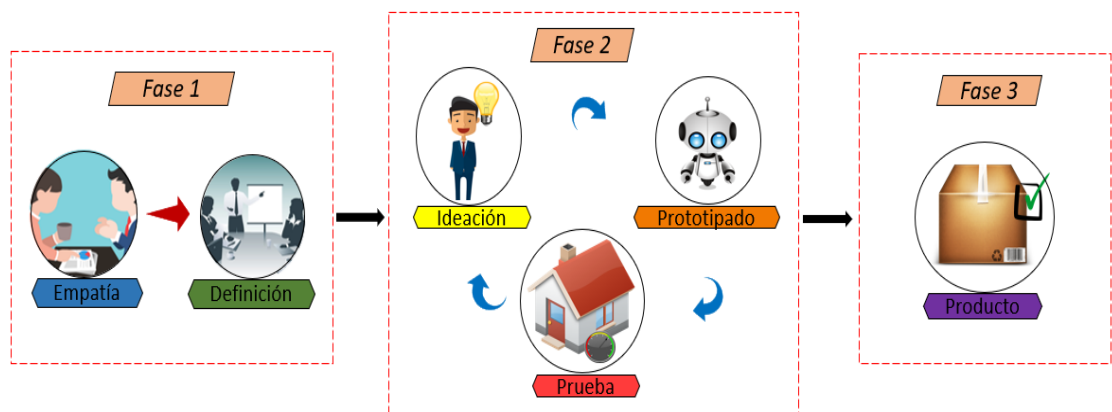
7.5. Planteamiento del Proyecto

Como aporte a la compañía en conocimientos del área electrónica y como aplicación de las metodologías empleadas por ellos, se realiza el siguiente planteamiento del proyecto con el fin de ser herramienta base en cuanto a la implementación de metodologías ágiles para ser utilizadas en la automatización de pruebas.

Se piensa en un proyecto que aporte conocimiento al grupo IES (Intelligent Engineering Services) de Accenture y genere una visión de la posible área de acción a trabajar dentro de la compañía. Se escoge la tecnología 'IoT' (Internet of Things) y para ello se realiza una mezcla entre la metodología 'Design Thinking' y la metodología 'Scrum'.

A continuación, se muestra una figura que contiene el esquema de trabajo del proyecto empleando las metodologías ágiles anteriormente mencionadas.

Figura 31. Metodología.



7.5.1. Fase I

Esta fase se dividió en dos etapas correspondientes a la metodología 'Design Thinking':

- Empatía:

En esta etapa se hizo un acercamiento constante con los miembros del 'Equipo IES' de Accenture para identificar qué áreas de conocimiento trabajan y cómo las emplean, de tal manera que me permitiera analizar una posible opción a trabajar.

- Definición:

Esta etapa se define el área de trabajo conocida como 'Domótica' que con base a la información obtenida en la etapa anterior permite mostrar una ampliación a sus conocimientos y una posibilidad de involucrarse en esta área.

7.5.2. Fase II

En esta fase se aplica la metodología 'Scrum' por medio de iteraciones incrementales que van generando un valor agregado en cada uno, se escogen las características o requerimientos principales a emplear en el proyecto en cada una de estas iteraciones. Esas características se desarrollan utilizando etapas de la metodología 'Design Thinking'.

- Ideación:

Como su nombre lo indica, en esta etapa surge una lluvia de ideas sobre las posibles funcionalidades o características que debe contener el proyecto escogiendo las más básicas y principales. Estas son: encendido/apagado de la luz, variación de la intensidad, temperatura, humedad y apertura/cierre de puerta.

- Prototipado:

En esta etapa se materializa cada una de las características correspondientes a cada iteración realizando el respectivo montaje o conexión de los sensores y/o dispositivos electrónicos (led's, sensor temperatura-humedad, servomotor, microcontrolador).

- Prueba:

En esta etapa se realiza la validación y verificación del correcto funcionamiento de la o las características escogidas para trabajar en cada iteración.

Nota: El desarrollo detallado de cada una de las iteraciones que componen esta fase se muestra más adelante en la sección 'Desarrollo de Proyecto'.

7.5.3. Fase III

En esta fase se culmina el proyecto con la última etapa.

- Producto:

En esta última etapa, se reúnen todas las características y funcionalidades agrupándolas ya en un único producto llamado 'producto final' como resultado de haber completado las etapas anteriores.

7.6. Desarrollo del Proyecto

Para la realización de este proyecto se realiza previamente una investigación sobre las posibles herramientas a utilizar, los componentes a desarrollar, la arquitectura a emplear y la metodología.

Por versatilidad y conectividad se escogió la plataforma de 'Ubidots' como interface de usuario para realizar los respectivos controles de variables escogidas previamente.

Seguidamente se realiza investigación sobre el tipo de hardware a trabajar teniendo como criterios: precios, versatilidad, manejo de varias variables analógicas y digitales, módulo wifi, tamaño y fácil acople con la plataforma de desarrollo 'Arduino'. El dispositivo que se selecciona es el módulo 'NodeMCU ESP8266' mostrado en la siguiente ilustración.

Figura 32. Módulo NodeMCU ESP8266.



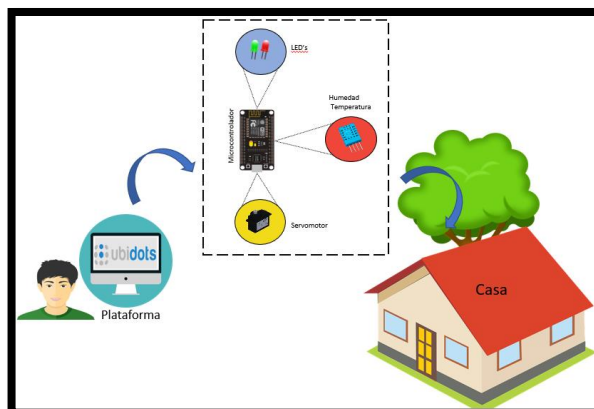
Tomado del link: <https://www.9arduino.com/product/68/nodemcu-v2-lua-based-esp8266-12e-nodemcu-v2>

El software que se escoge para programar el módulo anterior es el IDE de Arduino, debido a que se ha trabajado durante la carrera, se posee mayor conocimiento y familiarización en el uso de esta herramienta de programación.

7.6.1. Arquitectura

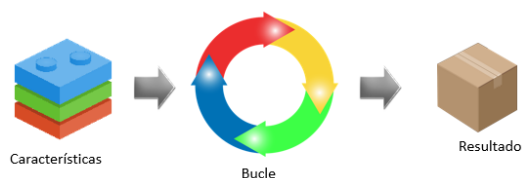
A continuación, se muestra en la siguiente ilustración la arquitectura a utilizar en el proyecto que va desde la plataforma visual del usuario hasta el dispositivo central encargado de ejecutar las diferentes acciones.

Figura 33. Arquitectura Planteada.



7.6.2. Ejecución

- Iteración I



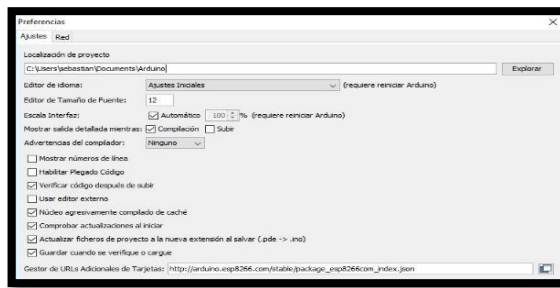
1) Ideación

- ✚ Uso IDE Arduino
- ✚ Instalación de Librería ESP8266
- ✚ Instalación de Librería de Ubidots
- ✚ Conexión Ubidots
- ✚ Encendido de LED
- ✚ Variación intensidad

2) Prototipado

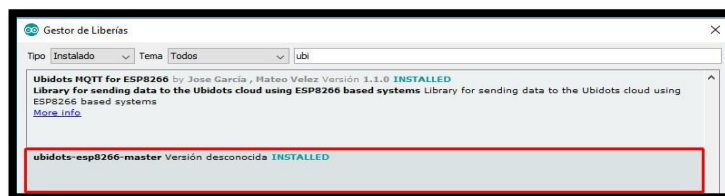
Para el manejo de la tarjeta ESP8266 o microcontrolador incluye la URL que se muestra en la figura 35, la cual contiene el 'package'. Este proceso se realiza seleccionando el menú 'Archivo' que está en la barra de menú del IDE de Arduino y luego seccionar la opción 'Preferencias'. Seguidamente, se abre una venta emergente en donde se ingresa 'https://Arduino.esp8266.com/stable/package_esp8266com_index.json' en el campo '**Gestor de URLs Adicionales de Tarjetas**' tal como se muestra en la siguiente ilustración.

Figura 34. Inclusión 'package' ESP8266.



Seguidamente se abre el **'Gesto de Librerías'** el cual se encuentra en el menú **'Programa'** en la opción de **'Incluir Librerías'** y luego el gestor. Una vez se esté en esa ventana se busca la librería **'ubidots-esp8266-master'** que contiene la librería tanto del módulo NodeMCU como la librería para la conexión con la plataforma Ubidots (ver figura 35).

Figura 35. Inclusión librerías NodeMCU y Ubidots.



A continuación se muestra el código implementado en la plataforma Arduino correspondiente a la primera iteración. Se observa la inclusión de la librería, la definición de variables fijas (token, red wifi y la contraseña) y el código principal para su funcionamiento.

```

***** /

//Ubidots
#include "UbidotsMicroESP8266.h"

/* *****
   Define Constants
   ***** */

#define intensity "5bac4927c03f97788c008065" // Put here your Ubidots
variable ID de Intensity
#define light "5bac4901c03f97788c008057" // Put here your Ubidots variable
ID de Light

#define TOKEN "A1E-tLEbEv7q7Wlr7XgGrW6JcCFTaBVk0t" /* Put here
your Ubidots TOKEN */
#define WIFISSID "Flia Lugo" /* Put here your Wi-Fi SSID */

```

```
#define PASSWORD "3115994570" /* Put here your Wi-Fi password  
3203255521 */
```

```
/******  
*****
```

Define Variables

```
*****  
*****/
```

```
int ledPin1 = 4; //D2  
int ledPin2 = 14; //D5  
int pos_anterior = 0;  
int pwm;// variar intensidad
```

```
/******  
*****
```

Auxiliar Functions

```
*****  
*****/
```

```
//Inicialización el cliente Ubidots  
Ubidots client(TOKEN);
```

```
/******  
*****
```

Proceso

```
*****  
*****/
```

```
void setup() {
```

```
    Serial.begin(9600);  
    client.wifiConnection(WIFISSID, PASSWORD);
```

```
    pinMode(ledPin1, OUTPUT);  
    pinMode(ledPin2, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    //Control On/Off  
    int value = client.getValue(light);  
    //Control Variable  
    int intensidad = client.getValue(intensity);
```

```
    //Analizar si se recibió el dato o no  
    if (value != ERROR_VALUE){  
        Serial.print(F("value obtained: "));  
        Serial.println(value);  
    }  
    else{  
        Serial.println(F("Error getting value"));  
    }  
    delay(250);
```

```

if (value == 1) {
  digitalWrite(ledPin2,HIGH);
}
else {
  digitalWrite(ledPin2,LOW);
}

if(intensidad >= 0 && intensidad <= 100)
{
  pwm = ((intensidad * 1024)/ 100);
  analogWrite(ledPin1,pwm);
}

Serial.print("Intensidad: ");
Serial.println(intensidad);
Serial.println();

delay(250);

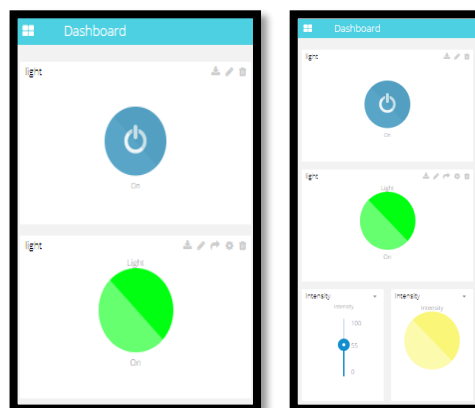
}
/*****
          Autor: Sebastian Augusto Baquero Peña
  *****/

```

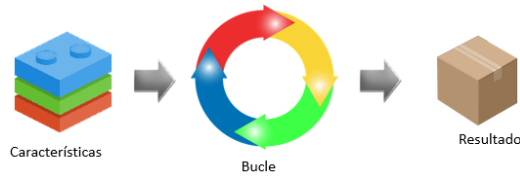
3) Prueba

En la figura 36 se muestra el Framework desarrollado en la plataforma de Ubidots para el encendido/apagado de un LED y la variación de la intensidad de otro LED.

Figura 36. Funcionamiento On/Off y Variación Intensidad.



- Iteración II



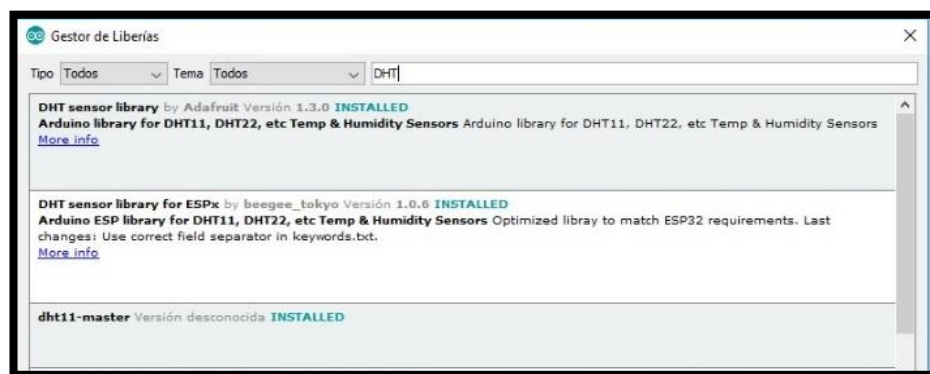
1) Ideación

- ✚ Instalación librería módulo DHT11 (sensor temperatura y presión)
- ✚ Funcionamiento en Arduino de solamente el sensor de temperatura y humedad con la plataforma Ubidots.

2) Prototipado

Se realiza la instalación de la librería del sensor de humedad y temperatura de la misma forma en que se hizo la instalación de la librería en la iteración anterior. El nombre de la librería a instalar se llama 'DHT sensor library for ESPx' o 'DHT sensor library' tal como se muestra en la figura 37.

Figura 37. Librería DHT11.



Seguidamente se procede a implementar el código para que permita obtener los datos de humedad y temperatura, para ello se inicia con la inclusión de la librería 'DHT.h', define el sensor a trabajar 'DHT11', se inicializa para su posterior control y ejecución.

```
/*  
  
  Include Libraries  
  */  
  
//Ubidots  
#include "UbidotsMicroESP8266.h"  
//Temperatura y Humedad  
#include <DHT.h>
```

```

/*****
    Define Constants
    *****/

#define intensity "5bac4927c03f97788c008065" // Put here your Ubidots
variable ID de Intensity
#define light "5bac4901c03f97788c008057" // Put here your Ubidots variable ID
de Light
#define temperature "5bac4700c03f977682515a47" // Put here your Ubidots
variable ID de temperatura
#define humidity "5bac4893c03f9777eccf446f" // Put here your Ubidots variable
ID de humedad

#define TOKEN "A1E-tLEbEv7q7Wlr7XgGrW6JjCFTaBVk0t" /* Put here your
Ubidots TOKEN */
#define WIFISSID "Flia Lugo" /* Put here your Wi-Fi SSID */
#define PASSWORD "3115994570" /* Put here your Wi-Fi password
3203255521 */

//Temperatura y Humedad
#define DHTPIN 5 //PIN D1 // Definimos el pin digital donde se conecta el sensor
// Dependiendo del tipo de sensor
#define DHTTYPE DHT11

/*****
    Define Variables
    *****/

int ledPin1 = 4; //D2
int ledPin2 = 14; //D5
int pos_anterior = 0;
int pwm;// variar intensidad

/*****
    Auxiliar Functions
    *****/

//Inicialización el cliente Ubidots
Ubidots client(TOKEN);
// Inicialización sensor DHT11
DHT dht(DHTPIN, DHTTYPE);

/*****
    Proceso
    *****/

void setup() {
    Serial.begin(9600);

```

```

client.wifiConnection(WIFISSID, PASSWORD);
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
//client.setDebug(true); // Uncomment this line to set DEBUG on

// Comenzamos el sensor DHT
dht.begin();
}

void loop() {

//Control On/Off
int value = client.getValue(light);
//Control Variable
int intensidad = client.getValue(intensity);
// Leemos la humedad relativa
float h = dht.readHumidity();
// Leemos la temperatura en grados celsius (por defecto)
float t = dht.readTemperature();
// Leemos la temperatura en grados Fahreheit
float f = dht.readTemperature(true);

//Analizar si se recibió el dato o no
if (value != ERROR_VALUE){
  Serial.print(F("value obtained: "));
  Serial.println(value);
}
else{
  Serial.println(F("Error getting value"));
}
delay(250);
// Comprobamos si ha habido algún error en la lectura
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Error obteniendo los datos del sensor DHT11");
  return;
}
if (h != 0 && t != 0){
  //Temperatura y Humedad
  client.add(temperature, t);
  client.add(humidity, h);
  client.sendAll(false);
}
if (value == 1) {
  digitalWrite(ledPin2,HIGH);
}
else {
  digitalWrite(ledPin2,LOW);
}
}

```

```

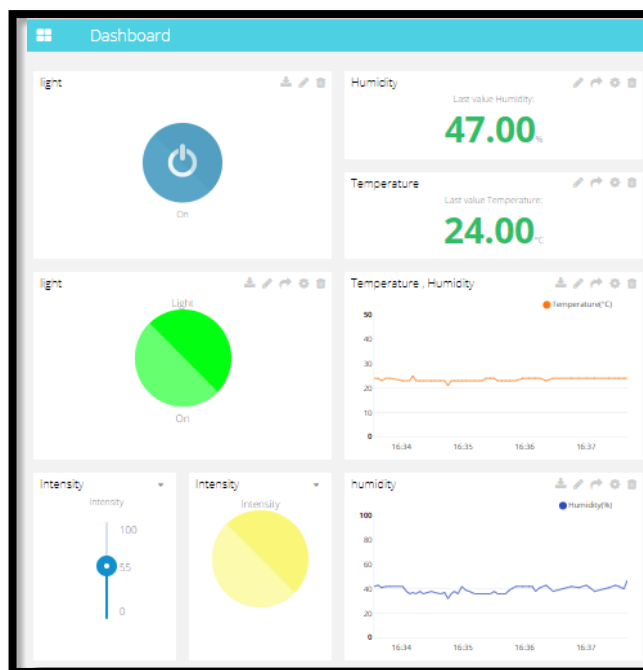
if(intensidad >= 0 && intensidad <= 100)
{
  pwm = ((intensidad * 1024)/ 100);
  analogWrite(ledPin1,pwm);
}
Serial.print("Humedad: ");
Serial.print(h);
Serial.println(" %\t");
Serial.print("Temperatura: ");
Serial.print(t);
Serial.print(" *C \n");
Serial.print("Intensidad: ");
Serial.println(intensidad);
Serial.print("PWM: ");
Serial.print(pwm);
Serial.println();
delay(250);
}
/*****
          Autor: Sebastian Augusto Baquero Peña
  * *****/

```

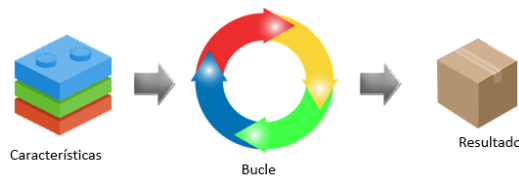
3) Prueba

En la figura 38 se evidencia el ‘framework’ en Ubidots con la implementación de los indicadores de humedad y temperatura en número y gráfica.

Figura 38. Humedad y Temperatura en Ubidots.



- Iteración III



1) Ideación

- 🔧 Instalación librería Servomotor
- 🔧 Funcionamiento físico del servomotor con la plataforma Ubitos.

2) Prototipado

Para el uso del servo motor se toma como plantilla base el ejemplo que trae predefinido Arduino, realizándole modificaciones para así poder tener control sobre este. Para ello se crean dos variables en el código para el posicionamiento del servo el cual va a servir para abrir o cerrar una puerta. El código desarrollado se muestra en la siguiente ilustración.

```
/*  
*****  
          Include Libraries  
*****/  
//Ubidots  
#include "UbidotsMicroESP8266.h"  
//Temperatura y Humedad  
#include <DHT.h>  
//Servo  
#include <Servo.h>  
/*  
*****  
          Define Constants  
*****/  
#define intensity  "5bac4927c03f97788c008065" // Put here your Ubidots  
variable ID de Intensity  
#define light  "5bac4901c03f97788c008057" // Put here your Ubidots variable ID  
de Light  
#define temperature  "5bac4700c03f977682515a47" // Put here your Ubidots  
variable ID de temperatura  
#define humidity  "5bac4893c03f9777eccf446f" // Put here your Ubidots variable  
ID de humedad  
#define servo  "5bac48ddc03f97788c008048" // Put here your Ubidots variable ID  
de servo  
  
#define TOKEN  "A1E-tLEbEv7q7Wlr7XgGrW6JjCFTaBVk0t" /* Put here your  
Ubidots TOKEN */  
#define WIFISSID  "Flia Lugo" /* Put here your Wi-Fi SSID */  
#define  PASSWORD  "3115994570" /* Put here your Wi-Fi password  
3203255521 */
```



```

//Temperatura y Humedad
#define DHTPIN 5 //PIN D1 // Definimos el pin digital donde se conecta el sensor
// Dependiendo del tipo de sensor
#define DHTTYPE DHT11
/*****

      Define Variables
*****/
int ledPin1 = 4; //D2
int ledPin2 = 14; //D5
int pos_anterior = 0;
int pwm;// variar intensidad
/*****

      Auxiliar Functions
*****/
//Inicialización el cliente Ubidots
Ubidots client(TOKEN);
// Inicialización sensor DHT11
DHT dht(DHTPIN, DHTTYPE);
//Inicialización Servo
Servo myservo;
/*****

      Proceso
*****/
void setup() {
  Serial.begin(9600);
  client.wifiConnection(WIFISSID, PASSWORD);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  //client.setDebug(true); // Uncomment this line to set DEBUG on

  // Comenzamos el sensor DHT
  dht.begin();
  // Servo
  myservo.attach(2); // D4 declarado como salida PWM
  myservo.write(0); //Ubicar el servo en la posición 0
}

void loop() {

  //Control On/Off
  int value = client.getValue(light);
  //Control Variable
  int intensidad = client.getValue(intensity);
  // Leemos la humedad relativa
  float h = dht.readHumidity();
  // Leemos la temperatura en grados celsius (por defecto)
  float t = dht.readTemperature();

```

```

// Leemos la temperatura en grados Fahreheit
float f = dht.readTemperature(true);
// Leemos la posición del servo
int pos_nueva = client.getValue(servo);
delay(100);

//Analizar si se recibió el dato o no
if (value != ERROR_VALUE){
  Serial.print(F("value obtained: "));
  Serial.println(value);
}
else{
  Serial.println(F("Error getting value"));
}
delay(250);
// Comprobamos si ha habido algún error en la lectura
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Error obteniendo los datos del sensor DHT11");
  return;
}

if (h != 0 && t != 0){
  //Temperatura y Humedad
  client.add(temperature, t);
  client.add(humidity, h);
  client.sendAll(false);
}
if (value == 1) {
  digitalWrite(ledPin2,HIGH);
}
else {
  digitalWrite(ledPin2,LOW);
}
if(intensidad >= 0 && intensidad <= 100)
{
  pwm = ((intensidad * 1024)/ 100);
  analogWrite(ledPin1,pwm);
}

Serial.print("Humedad: ");
Serial.print(h);
Serial.println(" %\t");
Serial.print("Temperatura: ");
Serial.print(t);
Serial.print(" *C \n");
Serial.print("Intensidad: ");
Serial.println(intensidad);
Serial.print("PWM: ");

```

```

Serial.print(pwm);
Serial.println();

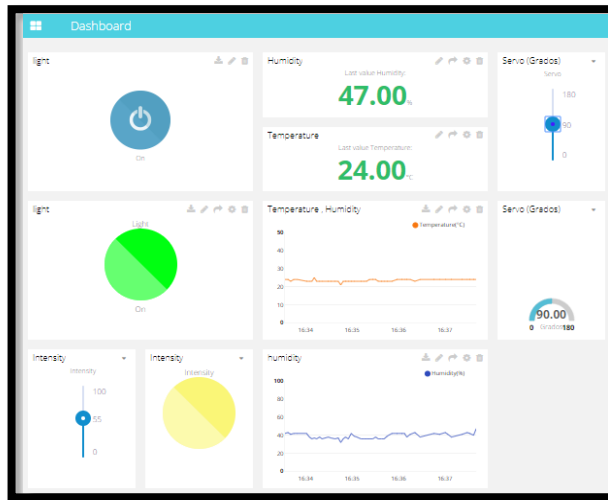
delay(250);
//Servo
Serial.println("-----");
  Serial.println("pos nueva: ");
  Serial.println(pos_nueva);
  Serial.println("pos anterior: ");
  Serial.println(pos_anterior);
  if(pos_nueva >= 0 && pos_nueva <= 180)
  {
    if(pos_nueva > pos_anterior)
    {
      for (int i = pos_anterior ; i<=pos_nueva; i++)
      {
        myservo.write(i);
        delay(15);
      }
    }
    else if(pos_nueva < pos_anterior)
    {
      for (int i = pos_anterior; i>=pos_nueva; i--)
      {
        myservo.write(i);
        delay(15);
      }
    }
    pos_anterior = pos_nueva;
  }
  else
  {
    Serial.println("Ingrese un valor mayor a 0");
  }
}
/*****
      Autor: Sebastian Augusto Baquero Peña
  * *****/

```

3) Prueba

Se realiza la adición del control e indicador de la posición en que se encuentra el servomotor el cual hace de rol de motor para abrir y cerrar una puerta. En la siguiente ilustración se muestra el 'framework' para el servo en Ubidots junto con los indicadores y controles anteriormente creados.

Figura 39. Control Servo en Ubidots.



7.6.3. Producto Final

Una vez completadas satisfactoriamente las iteraciones anteriores, se realiza la maqueta del prototipo.

En las figuras 40 y 41, se muestra el diseño del prototipo que se construyó donde cuenta con una abertura en la parte superior con el objetivo que el sensor de temperatura y humedad obtenga una mejor medición, también cuenta con dos led's: el primero (rojo) corresponde al control ON/OFF y el segundo (amarillo) al control o variación de intensidad y finalmente en la parte frontal de la 'casa' se muestra una puerta la cual es controlada por un servomotor para hacer la apertura o cierre de ésta.

Figura 40. Prototipo terminado.

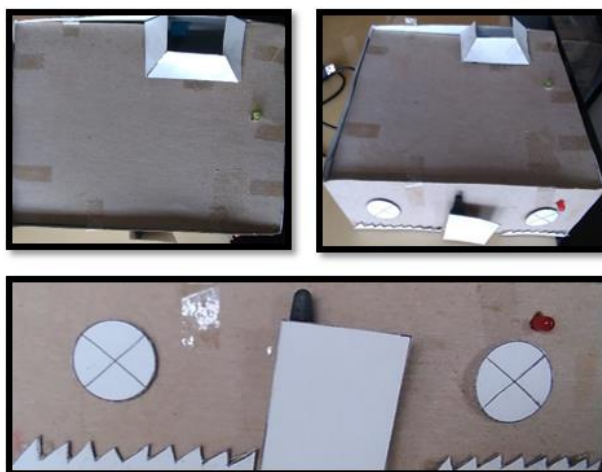
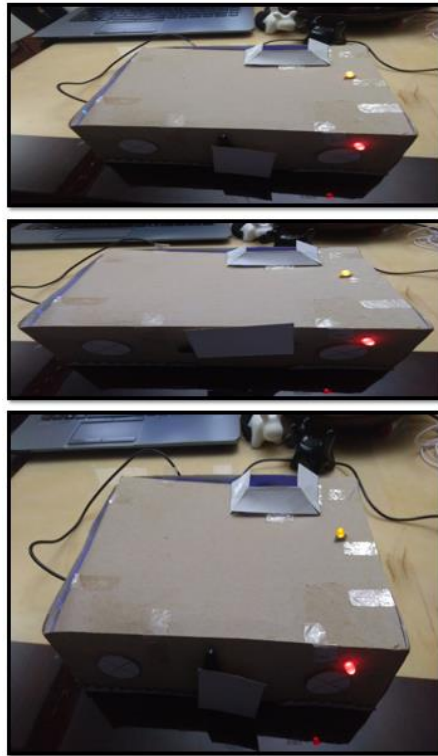


Figura 41. Prototipo en funcionamiento.



8. DESARROLLO DE AUTOMATIZACIÓN DE PRUEBAS

Para el desarrollo de la automatización de pruebas se escogió la herramienta de desarrollo de Visual Studio debido a que es una herramienta que facilita la automatización de aplicaciones web mediante el uso de varias librerías o componentes 'NuGets' que permite dar valor agregado a las pruebas.

Para el proceso de automatización de uno de los módulos de la plataforma de principal de Claro, se realizó de la siguiente manera:

Figura 42. Estructura de Trabajo.

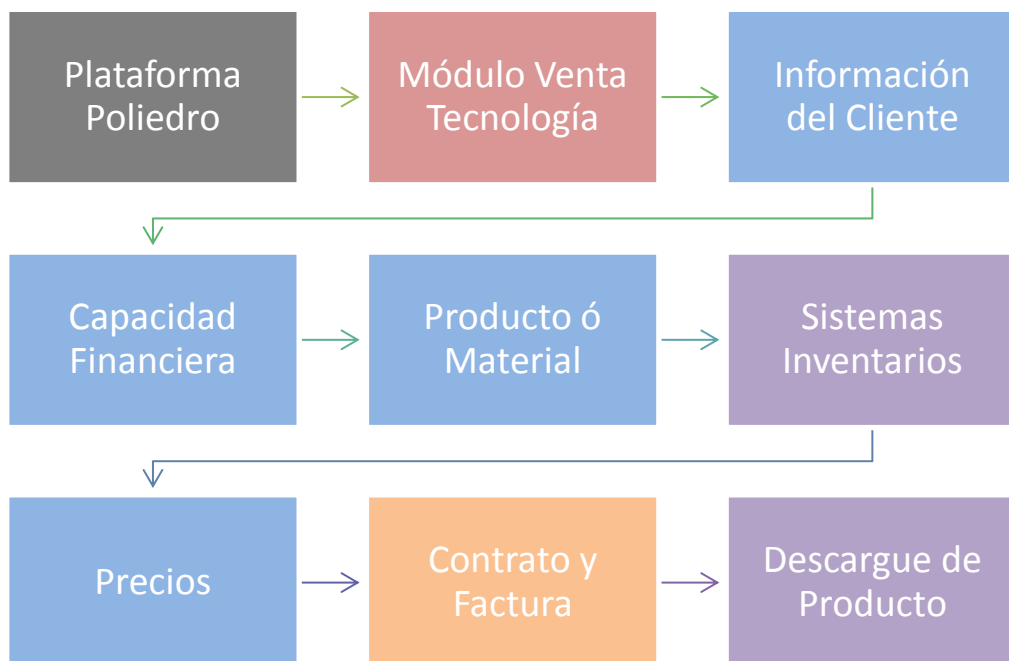


8.1. Descubrimiento

Para el funcionamiento de la plataforma y de las diferentes pruebas a automatizar, se realiza el reconocimiento, obteniendo el siguiente diagrama con el funcionamiento básico perteneciente a todos los escenarios del cual se parte para la planeación del desarrollo.

En la figura 43, se muestra el flujo básico de la aplicación automatizada comenzando con el ingreso a la plataforma 'Poliedro' con el usuario y contraseña autorizado, después se ingresa al módulo de 'Venta Tecnología' ingresando los datos correspondientes al cliente, seguidamente se valida la información del cliente y capacidad de financiamiento, una vez cumplido con este proceso se selecciona el producto, se verifican su precio, paralelamente se realiza el ajuste en inventarios para luego generar el contrato y la factura a cancelar, se da salida al producto en inventarios y se finaliza la venta.

Figura 43. Flujo básico pruebas.

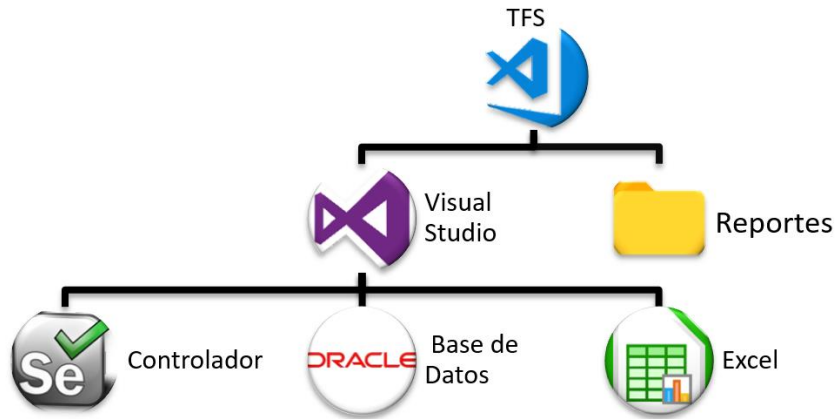


8.2. Planeación

En esta etapa se identificaron los escenarios a intervenir con su respectiva información a utilizar. Posteriormente, se realizó una estructura de los componentes que se deben involucrar para desarrollar las pruebas de automatización más completas. En la figura 44, se observa como orquestador y repositorio a la herramienta TFS, quien se encarga de alojar el proyecto desarrollado en Visual Studio y a su vez este genera un reporte al lanzar las pruebas automatizadas. Finalmente, el manejo de la información de los escenarios se almacena en un archivo de Excel, la configuración en base de

datos Oracle y el controlador de navegadores se añaden al proyecto en Visual Studio.

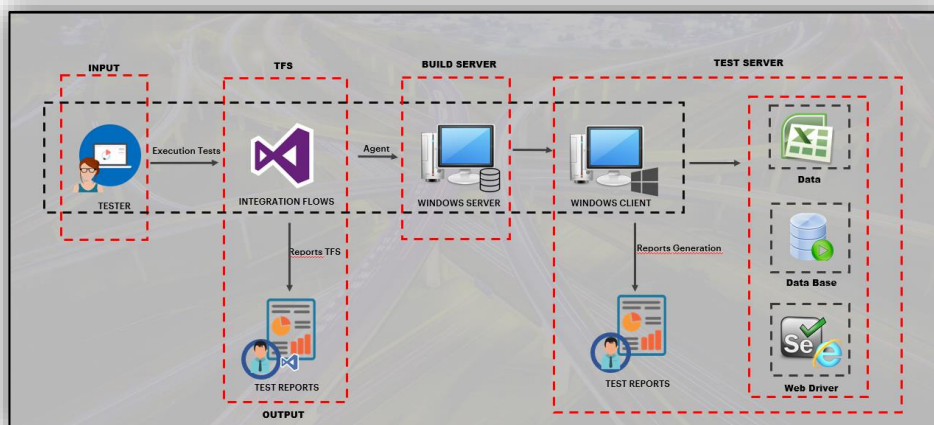
Figura 44. Estructura componentes automatización.



Arquitectura:

En la siguiente imagen se muestra la arquitectura básica que fue empleada para el proyecto para la automatización de pruebas partiendo del 'Tester' (persona encargada de lanzar o ejecutar las pruebas), pasando por el orquestador TFS, seguidamente por el servidor de construcción del proyecto hasta finalizar con la ejecución de la solución desarrollada.

Figura 45. Arquitectura de automatización de pruebas.



8.3. Desarrollo

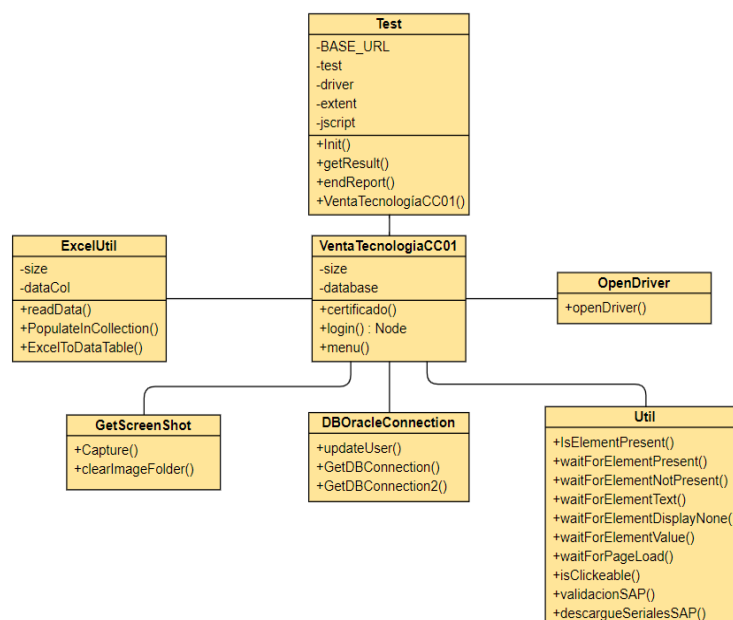
En esta etapa se realiza la creación del archivo de Excel con los campos importantes a utilizar en cada uno de los escenarios de pruebas el cual serán tomados gracias a la librería 'ExcelDataReader'. En la siguiente figura se muestran algunos de los campos empleados.

Figura 46. Archivo Excel.

A	B	C	D	E	F	G	H	I	J	K	L	M
Ambiente	Navegador	Usuario	Contraseña	Tipo_Documento	Documento	MIN	Tipo_Pago	Tipo_Cuenta	Tipo_Equipo	Fabricante1	Producto1	Fabricante2

Se realizó el desarrollo de cada uno de los escenarios de flujos contemplados para el proyecto utilizan las clases que se muestran en la figura 47.

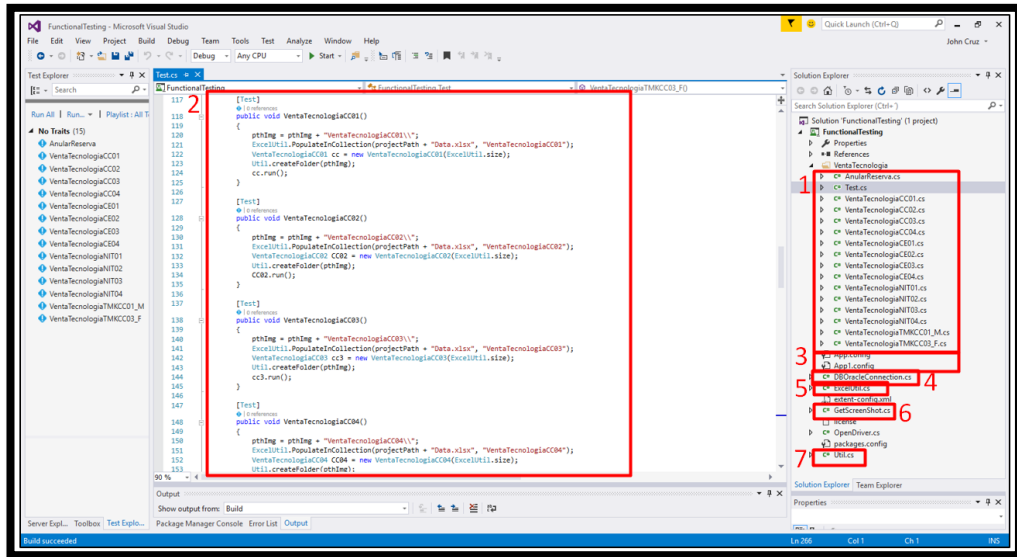
Figura 47. Diagrama de clases.



1. Escenarios automatizados: Comprende los escenarios que fueron automatizados en el proyecto.
2. Subcontenido escenarios: Pre-condiciones para cada escenario.
3. Archivos Configuración – ‘Properties’: Contiene variables de confidencialidad.
4. Base de Datos: Conexión y ejecución de ‘querys’ a la base de datos Oracle.
5. Archivo Excel: Lectura de las hojas y campos de los escenarios en el formato de Excel.
6. Pantallazos: Realiza la toma de pantallazos durante la ejecución de cada escenario automatizado.
7. Utilidades: Contiene los métodos a utilizar para la automatización de la plataforma web.

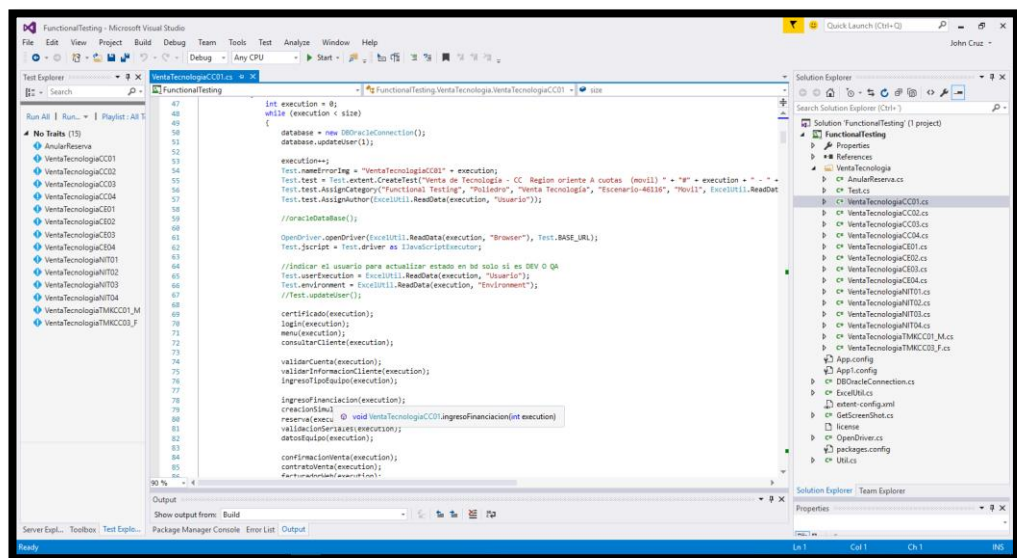
En la figura 48 se muestra parte del contenido de los diferentes escenarios de pruebas alojado en una clase principal, la cual se encarga de llamar a cada una de ellas y ejecutarla.

Figura 48. Desarrollo - Parte I.



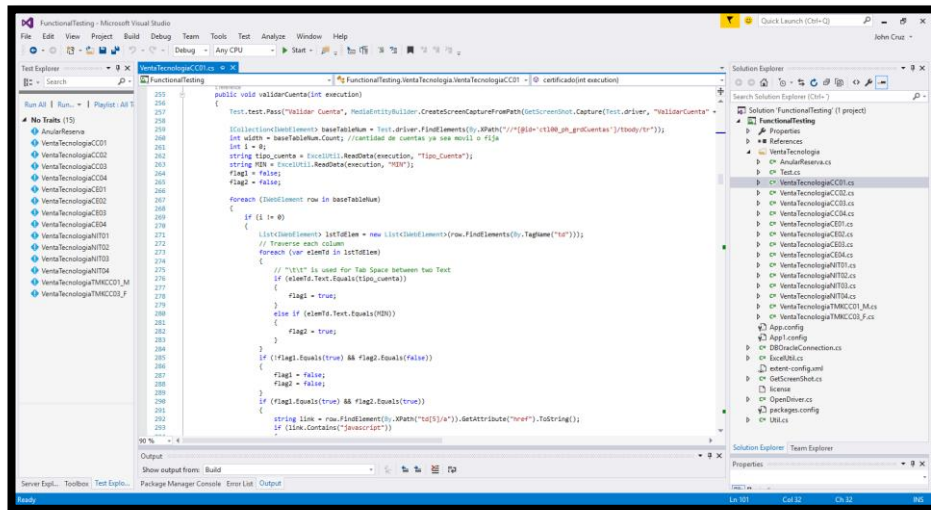
En la figura 49 se muestra parte del contenido que contiene un escenario y a su vez los diferentes métodos involucrados dentro de todo el proceso de automatización de un escenario.

Figura 49. Desarrollo - Parte II.



En la Figura 50, se muestra más en detalle el contenido de uno de los métodos pertenecientes al escenario, esto con el fin de dar una idea de la estructura que generalmente posee cada uno de los métodos y los elementos involucrados.

Figura 50. Desarrollo - Parte III.



8.4. Pruebas

En esta etapa se realiza la ejecución de las pruebas.

Figura 51. Ciclo Automatización de Pruebas.



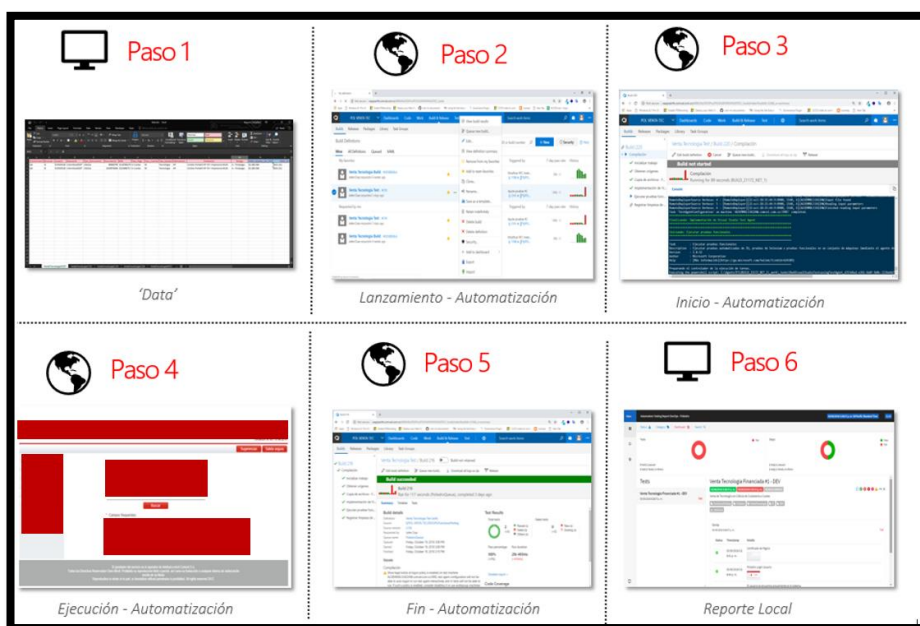
1. Ingreso Navegador: El controlador accede al navegador 'Internet Explorer' a la ruta destino de la plataforma en ambiente de pruebas accediendo al certificado de seguridad.
2. Plataforma Poliedro – Parte I: Ingreso a la plataforma con el usuario y contraseña, datos del cliente, selección del producto, modalidad de pago, generación de reserva.

3. Sistema de Inventarios – Parte I: Validación material asignado al cliente.
4. Plataforma Poliedro – Parte II: Ingreso monto a cancelar, autorización del cliente, generación contrato, generación factura.
5. Sistema de Inventarios – Parte II: Validación descargue del producto de acuerdo con la reserva creada.
6. Plataforma Poliedro – Parte III: Verificación de los procesos de la venta ejecutados con éxito. *Nota: Una vez se termine una prueba se continua con la siguiente hasta terminar todas las pruebas.*

8.5. Implementación

En esta etapa se muestra el paso a paso de la implementación de las pruebas automatizadas desde la configuración de la información pasando por el proceso de lanzamiento en TFS seguido del recorrido sobre la plataforma hasta llegar a los reportes finales (ver figura 52).

Figura 52. Implementación.



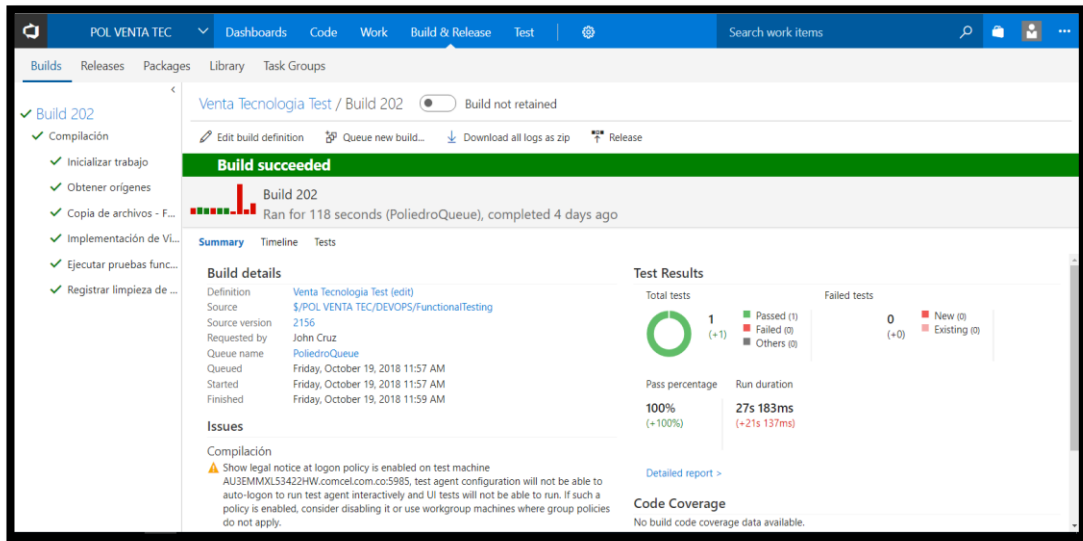
- Paso 1: Diligencias los campos en el Excel para el escenario de prueba a automatizar.
- Paso 2: Ingresa a la herramienta Visual Studio Team Foundation Server donde se encuentra configurada el lanzamiento de las pruebas automatizadas y se realiza su ejecución.
- Paso 3: Se realiza la inicialización de trabajo, obtención de orígenes, copia de archivo, implementación, ejecución de pruebas funcionales contenidas en el proceso de compilación.
- Paso 4: Automatización plataforma de poliedro.

- Paso 5: Registro de limpieza, mostrando el resultado de la ejecución de la automatización.
- Paso 6: Reporte local con más detalles de las pruebas automatizadas.

9. RESULTADOS

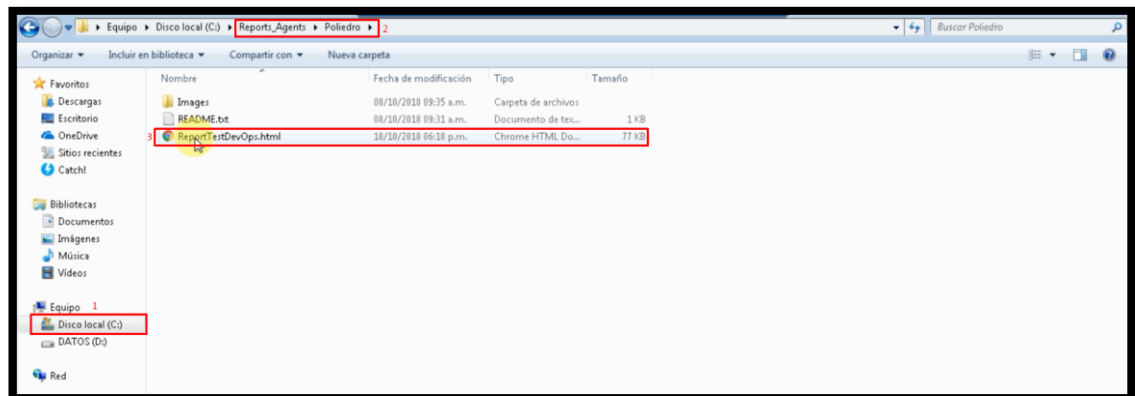
A continuación, se muestra la pantalla del resultado obtenido en la herramienta TFS una vez se ha finalizado la ejecución del escenario automatizado.

Figura 53. Reporte final TFS.



La siguiente imagen, muestra el directorio donde se alojan tres archivos: carpeta que contiene los pantallazos de cada una de las partes que componen el flujo completo del escenario, un archivo 'README' de formato texto y un reporte en 'html' generado por la librería 'ExtentReport' del proyecto de Visual Studio.

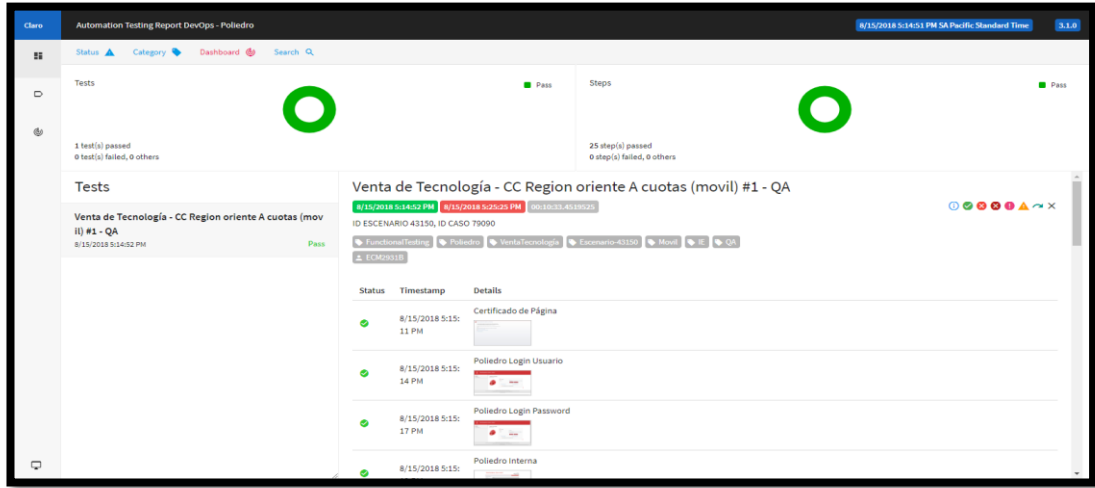
Figura 54. Reporte Local.



En la figura 55, se muestra el reporte creado con la librería 'ExtentReport' incluida en el proyecto. En este reporte se evidencia el nombre de la prueba ejecutada, los pasos realizados, pruebas realizadas y diagramas. Estos resultados pueden

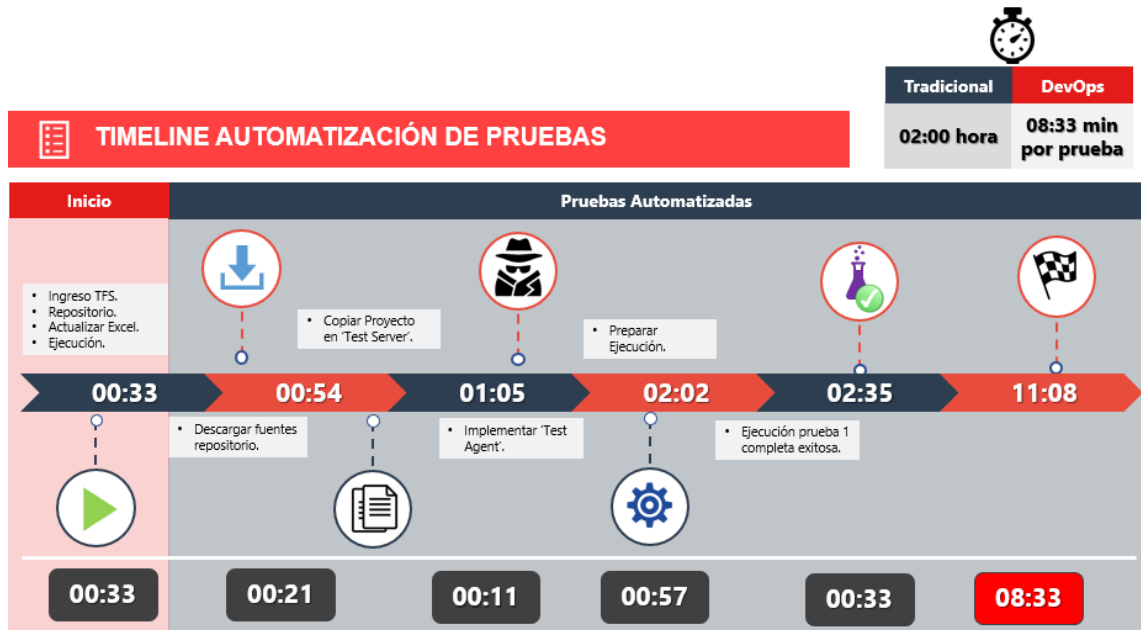
ser exitosos o erróneos tanto para los pasos como para las pruebas y ser visualizados en los diagramas dona del resultado global.

Figura 55. Reporte formato 'HTML'.



Con base a un análisis experimental comparando el tiempo de ejecución de una prueba manual con respecto a una prueba automatizada se evidencia en la siguiente imagen una reducción muy significativa. Esta reducción se debe a la toma de pantallazos mientras se está ejecutando la prueba y a la previa configuración de la información a utilizar.

Figura 56. Tiempos de una prueba automatizada.



Apoyo Documentación

Actividades:

- *Revisión de documentos de funcionamiento de la empresa y su modelo de negocio.*
- *Realización de diagramas de flujo y secuencia para cada uno de los escenarios de la plataforma.*
- *Realización de un manual de la descripción funcional de un requerimiento por parte del cliente en el cual se involucró el diseño de la arquitectura solución, modelo de datos y sus componentes, diagramas de secuencias, etc.*

Metodología

- *Leer documentación para entender el modelo de negocio, funcionamiento del cliente y los requerimientos de la aplicación.*
- *Revisión conceptual del correcto diseño de diagramas de secuencia y flujo en desarrollo de software.*
- *Levantamiento de solicitudes por parte del cliente para realizar modificaciones a la documentación.*



Fortaleza

- *Buen manejo de la comunicación con el cliente.*
- *Disciplina en la investigación y ejecución del diseño de diagramas.*
- *Adaptación al cambio.*
- *Versatilidad en el manejo de diferentes herramientas como word, excel y visio.*
- *Participación activa con los miembros del equipo de trabajo.*

Puntos a Mejorar

- *Para un excelente entrega y una mejora en las versiones de la documentación se realizó varias reuniones con la persona encargada de la revisión con el fin de aclarar varias inquietudes, ajustar el documento de acuerdo a lo solicitado por el cliente.*
- *Para la mejora en la realización de diagramas de secuencia se pidió acompañamiento y asesoría a un compañero quien tiene una gran experiencia en el tema.*

Apoyo Testing

Actividades:

- *Monitoreo y análisis del correcto funcionamiento de la aplicación de acuerdo a los escenarios de pruebas.*
- *Reporte de errores funcionales de los escenarios.*
- *Manejo y control de Bases de Datos como soporte en la ejecuciones de los diferentes escenarios de la plataforma.*
- *Uso de herramienta de pruebas de aplicaciones SoapUI.*

Metodología

- *Para el monitorio y análisis del funcionamiento se hizo previo una revisión del la documentación.*
- *Utilizando un formato para enviar los resportes y escalarlos al área respectiva.*
- *En el manejo de la base de datos se recibe una capacitación por parte de un compañero del equipo de fábrica.*
- *Inducción en el manejo de la herramienta SoapUI*



Fortalezas

- *Adaptación rápida en el manejo de la plataforma y sus diferentes escenarios de pruebas.*
- *Aprendizaje rápido en el uso y manejo de base de datos.*
- *Ejecuciones rápidas de los escenarios de pruebas como validación de su correcto funcionamiento.*

Puntos a Mejorar

- *Tener en cuenta las evidencias de cada una de las pruebas ejecutadas como material de soporte.*

Nota: No se evidencia información más detallada por políticas de confidencialidad entre ambas partes.

10. CONCLUSIONES

Mediante la investigación se pudo conocer la gran variedad de metodologías ágiles y su principal enfoque de trabajo especialmente las más utilizadas por Accenture siendo dos de ellas la metodología Scrum y la metodología DevOps.

El diseño y la implementación de un prototipo de Domótica permitió la aplicación de algunas de las metodologías ágiles empleadas por Accenture logrando su realización en un tiempo corto sirviendo como prueba de concepto para los diferentes miembros de grupo IES de Accenture así mismo cómo prueba de aplicación de conocimientos técnicos y formativos adquiridos en el proceso académico de formación como profesional.

Se selecciona la herramienta 'Selenium' trabajada en Visual Studio como una herramienta óptima y versátil a la hora de su construcción, ejecución e implementación siendo ésta una de las primeras en el escalafón de las herramientas más utilizadas para la automatización de pruebas.

Realizando una comparación experimental entre el tiempo de ejecución de una prueba manual contra una prueba automatizada, se obtuvo una reducción muy significativa del tiempo permitiendo abarcar una gran cantidad de escenarios de pruebas en un menor tiempo, siendo éste de gran importancia para la validación de un producto.

Gracias al conocimiento de las metodologías se pudo aplicar para un caso de la tecnología del Internet de las Cosas, así mismo se logró realizar unas buenas prácticas en el proceso de automatización de una aplicación de pruebas evidenciando al cliente las grandes ventajas en cuanto a tiempo que éstas permiten lograr.

La práctica permitió la maduración y la aplicación de todas las herramientas adquiridas a lo largo del proceso de formación en la universidad abarcando habilidades técnicas, aptitudes tanto a nivel académico como social e interpersonal.

11. BIBLIOGRAFÍA

- [1] Accenture, «Acerca de Accenture,» Accenture, 2018. [En línea]. Available: <https://www.accenture.com/co-es/company>.
- [2] América Móvil, «Introducción,» América Móvil, 2018. [En línea]. Available: <http://www.americamovil.com/Spanish/acerca-de-nosotros/nuestra-empresa/default.aspx>.
- [3] Claro, «Sobre nosotros,» América Móvil, 2018. [En línea]. Available: <https://www.claro.com/paises/quienes-somos.html>.
- [4] Comisión de Regulación de Comunicaciones, «Reporte de Industria Sector TIC,» Comisión de Regulación de Comunicaciones, 2017.
- [5] Super Intendencia de Industria y Comercio, «Información Institucional,» Super Intendencia de Industria y Comercio, [En línea]. Available: <http://www.sic.gov.co/objetivos-y-funciones>.
- [6] MinTIC, «Acerca del MinTIC,» MinTIC, 9 Marzo 2018. [En línea]. Available: <https://www.mintic.gov.co/portal/604/w3-propertyvalue-540.html>.
- [7] Comisión de Regulación de Comunicaciones, «Funciones y Deberes CRC,» Comisión de Regulación de Comunicaciones, [En línea]. Available: <https://www.crcm.gov.co/es/pagina/funciones-y-deberes-de-la-crc>.
- [8] MinTIC, «Autoridad Nacional de Televisión,» MinTIC, 12 Febrero 2014. [En línea]. Available: <https://www.mintic.gov.co/portal/604/w3-propertyvalue-6181.html>.
- [9] Superintendencia de Industria y Comercio, «Superintendencia de Servicios Públicos domiciliarios,» Red Nacional de Protección al Consumidor, [En línea]. Available: http://www.redconsumidor.gov.co/publicaciones/superintendencia_de_servicios_publicos_domiciliarios_pub.
- [10] Linchpin SEO, «A Beginners Guide To Understanding The Agile Method,» 1 January 2013. [En línea]. Available: <https://linchpinseo.com/the-agile-method/>.
- [11] S. Ashmore y K. Runyan, «The History and Value of Agile Software Development, » de Introduction to Agile Methods, vol. 1, Addison-Wesley, 2014, p. 336.
- [12] P. Abrahamsson, O. Salo, J. Ronkainen y J. Wartsa, «Agile Software Development Methods: Review and Analysis,» *ResearchGate*, p. 107, January 2002.
- [13] ING, «El auge de las metodologías ágiles: así ha cambiado nuestra forma de trabajar,» ING People in Progress, 28 Marzo 2018. [En línea]. Available: <https://www.ennaranja.com/agile/el-auge-de-las-metodologias-agiles/>.
- [14] OBS Business School, «Principales ventajas y limitaciones de las metodologías ágiles,» OBS Business School, 2018. [En línea]. Available: <https://www.obs-edu.com/int/blog-project-management/metodologia-agile/principales-ventajas-y-limitaciones-de-las-metodologias-agiles>.
- [15] Solvingad hoc, «TIPOS DE METODOLOGÍA ÁGIL Y SU APLICACIÓN EMPRESARIAL,» Solving Ad Hoc, 26 Junio 2017. [En línea]. Available: <https://solvingad hoc.com/tipos-metodologia-agil/>.
- [16] J. Mas y C. Mateu, «Introduction to Web Application Development,» de Introduction to Web Application Development, Barcelona, Free Technology Academy, 2010, p. 377.
- [17] D. Nations, «What Exactly is a Web Application,» Lifewire, 22 February 2018. [En línea]. Available: <https://www.lifewire.com/what-is-a-web-application-3486637>.

- [18] Computer Hope, «Web Application,» Computer , 26 April 2017. [En línea]. Available: <https://www.computerhope.com/jargon/w/web-application.htm>.
- [19] A. Ndegwa, «What is a Web Application,» STACKPATH, 31 May 2016. [En línea]. Available: <https://www.maxcdn.com/one/visual-glossary/web-application/>.
- [20] S. Banerjee, «6 Phases to Complete a Web Development Project Successfully,» RSWEBSOLS, 16 February 2016. [En línea]. Available: <https://www.rwebsols.com/tutorials/web-design/web-development-project-phases>.
- [21] A. Smith, «THE BEST WEB FRONTEND DEVELOPMENT TOOLS IN 2018,» Mobile Jazz, 22 November 2017. [En línea]. Available: <https://mobilejazz.com/blog/the-best-web-frontend-development-tools-in-2018/>.
- [22] S. Sharma y V. Sharma, «STUDY AND ANALYSIS OF AUTOMATION TESTING TECHNIQUES,» *Journal of Global Research in Computer Science*, vol. 3, p. 8, 2012.
- [23] R. Kalindri, «Automatizar las pruebas funcionales,» *Ciencias Computacionales y la Ingeniería de Software*, vol. 4, p. 2, 2014.
- [24] Toka Systems Development, «Servicios,» TOKA SYSTEMS - Testing On Knowledge Applied, 2013. [En línea]. Available: <http://toka-solutions.com/servicios.html>.
- [25] Panel Testing - Centro de Excelencia, «Software QA – ¿Cuáles son los tipos de pruebas software?,» Panel, 11 Febrero 2015. [En línea]. Available: <https://www.panel.es/blog/software-qa-cuales-son-los-tipos-de-pruebas-software/>.
- [26] P. Ritchie, *Practical Microsoft Visual Studio 2015*, Arizona: Apress, 2016, p. 209.
- [27] S. St. Jean, D. Brady, E. Blankenship, M. Woodward y G. Holliday, *Professional Team Foundation Server 2013*, Indianapolis: Wrox, 2014.
- [28] Microsoft, «Team Foundation Sever,» Microsoft, 2018. [En línea]. Available: <https://visualstudio.microsoft.com/tfs/?rr=https%3A%2F%2Fwww.google.com.co%2F>.
- [29] M. E. Moreira, M. Lester y S. Holzner, *Agile for Dummies*, C. Technologies, Ed., Indianapolis, Indiana: Wiley Publishing, Inc., 2010.
- [30] K. S. Rubin, *Essential Scrum: A Practical Guide to The Most Popular Agile Process*, Michigan, Ann Arbor: Addison Wesley, 2012.
- [31] Forbes, «The Benefits of Using Agile Software Development, » Forbes, 9 May 2016. [En línea]. Available: <https://www.forbes.com/sites/forbestechcouncil/2016/05/09/the-benefits-of-using-agile-software-development/#5f64db99b0f8>.
- [32] S. Sharma y B. Coyne, *DevOps for DUMMIES*, vol. 2, John Wiley & Sons, Inc., 2015.
- [33] Tecnologías Informacion, «DevOps Conceptos Básicos,» Tecnologías Informacion, [En línea]. Available: <https://www.tecnologias-informacion.com/devops.html#>.
- [34] R. Buyya y A. Vahid Dastjerdi, «Internet of Things: Principles and Paradimgs,» de *Internet of Things*, A. Invernizzi, Ed., Cambridge, Massachusetts: Morgan Kaufmann, 2016, p. 354.
- [35] Arduino, «Wifi Web Server, » Arduino, 2018. [En línea]. Available: <https://www.arduino.cc/en/Tutorial/WiFiWebServer>.
- [36] Jerry, «Arduino webserver:faster alternative to "indexof" for parsing GET requests,» Arduino StackExchange, 12 Julio 2017. [En línea]. Available: <https://arduino.stackexchange.com/questions/13388/arduino-webserver-faster-alternative-to-indexof-for-parsing-get-request>.
- [37] Á. Cárdenas, «Encender una bombilla con un módulo wifi ESP8266,» Semotic, 30 abril 2015. [En línea]. Available: <https://secmotic.com/encender-bombilla-con-un-modulo-wifi-esp8266-arduino>.
- [38] M. C. Hernandez, «Connect a NodeMCU ESP8266 to Ubidots over HTTP,» Ubidots, 2018. [En línea]. Available: <https://help.ubidots.com/connect-your-devices/connect-a-nodemcu-esp8266-to-ubidots-over-http>.

[39] Instructables, «MIDIENDO TEMPERATURA CON ESP8266 Y DHT11,» The Invento's House, 2018. [En línea]. Available: <https://www.instructables.com/id/Midiendo-temperatura-con-ESP8266-y-DHT11>.

[40] L. D. V. Hernández, «Cómo utilizar el sensor DHT11 para medir la temperatura y humedad con Arduino.,» Programarfacil, 2017. [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino>.