



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PRÁCTICA PROFESIONAL

INFORME PRÁCTICA PROFESIONAL, SAN SERVICES S. DE R.L

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

21411030 JOSÉ ANTONIO MEJÍA PERDOMO

ASESOR: ING. CÉSAR ORELLANA

CAMPUS SAN PEDRO SULA

06 DE MAYO DE 2020

ÍNDICE

I. Introducción	1
II. Generalidades de la empresa.....	2
2.1 Descripción de la Empresa.....	2
2.2 Descripción del Departamento	2
2.3 Objetivos.....	3
2.3.1 Objetivo General.....	3
2.3.2 Objetivos Específicos	3
III. Marco Teórico	4
3.1 Rubro del proyecto.....	4
3.1.1 CONTROL DE CALIDAD (QUALITY ASSURANCE).....	4
3.2 Metodología de Trabajo	6
3.2.1 DESARROLLO AGILE.....	6
3.2.2 SCRUM.....	8
3.2.3 KANBAN.....	10
3.3 Tecnologías utilizadas.....	11
3.3.1 JAVASCRIPT	11
3.3.2 NODEJS.....	14
3.3.3 TRELLO	14
3.3.4 BABEL.....	14
3.3.5 WEBDRIVERIO	15

3.3.6 MOCHA JS	20
3.3.7 CHAI JS.....	21
3.3.8 SELENOID.....	22
3.3.9 GIT.....	23
3.3.10 RUNDECK	27
IV. Descripción del trabajo realizado.....	28
4.1 Proceso de capacitación	28
4.1.1 Tecnologías utilizadas en los proyectos	28
4.2 Proyectos como parte del equipo de desarrollo.....	34
4.2.1 QA Contents.....	35
4.2.2 QA QOBE	69
V. Conclusiones.....	78
VI. Recomendaciones	79
VII. Bibliografías	80

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Roles, artefactos y eventos principales de Scrum.....	9
Ilustración 2: Límites WIP.....	11
Ilustración 3: Conversión automática en JS.....	13
Ilustración 4: Configuración de specs.....	16
Ilustración 5: Configuración de exclude	16
Ilustración 6: Configuración de capabilities	17
Ilustración 7: Configuración de services y framework.....	17
Ilustración 8: Configuración de reporters	18
Ilustración 9: Configuración de user, key y región	18
Ilustración 10: Ejemplo de un hook.....	19
Ilustración 11: Estructura básica de un page object	19
Ilustración 12: Ejemplo del uso de it y describe	20
Ilustración 13: Reporte generado por Mochawesome.....	21
Ilustración 14: Ejemplo de expect	22
Ilustración 15: Ejemplo de should.....	22
Ilustración 16: Interfaz gráfica de Selenoid	23
Ilustración 17: Directorio de trabajo, área de preparación y el directorio Git.....	25
Ilustración 18: Inicializando un nuevo repositorio.....	25
Ilustración 19: Seguimiento de archivos.....	26
Ilustración 20: Confirmando cambios.....	26
Ilustración 21: Subiendo cambios al repositorio remoto.....	26
Ilustración 22: Descargando cambios de un repositorio remoto	26

Ilustración 23: Clonando un repositorio.....	27
Ilustración 24: Interfaz visual de Rundeck	27
Ilustración 25: Estructura de los proyectos.....	30
Ilustración 26: Estructura de un spec.....	32
Ilustración 27: Estructura del archivo de configuración	33
Ilustración 28: Paginas de Sandals Inclusions	36
Ilustración 29: Paginas de Sandals Weddings.....	36
Ilustración 30: Resto de sitios de Sandals.....	37
Ilustración 31: Paginas de Beaches Inclusions.....	37
Ilustración 32: Paginas de Beaches For Everyone	38
Ilustración 33: Paginas de Beaches Weddings.....	38
Ilustración 34: Resto de páginas de Beaches.....	39
Ilustración 35: Selectores de las imágenes de la página de Accommodation.....	40
Ilustración 36: Selectores de los enlaces de la página de Accommodation.....	41
Ilustración 37: Función para cerrar popup.....	41
Ilustración 38: Step object para los enlaces	42
Ilustración 39: Función para realizar request.....	43
Ilustración 40: Step object para manipular las imágenes	44
Ilustración 41: Step object para manipular los botones.....	45
Ilustración 42: Spec de Sandals Accommodation.....	47
Ilustración 43: Selectores para la página de Butler Services.....	47
Ilustración 44: Page object de la página Butler Services.....	48
Ilustración 45: Spec de la página Butler Services	48

Ilustración 46:Page objects de Best Beaches y Sandals Club	49
Ilustración 47: Step object para manipular todo tipo de step object	50
Ilustración 48: Step object para la manipulación de los popups	51
Ilustración 49: Src de una lazy-image que no a termina de cargar.....	52
Ilustración 50: Step object que manipula los dropdowns	53
Ilustración 51: Step object que manipula los checkboxes	54
Ilustración 52: Page object de Weddings Overview.....	58
Ilustración 53: Spec de Weddings Overview.....	58
Ilustración 54: Step object para manipular los precios.....	60
Ilustración 55: Función para tomar screenshots 1	63
Ilustración 56: Función para tomar screenshots 2	63
Ilustración 57: Step object para detectar el estado 404.....	64
Ilustración 58: Función para generar reportes	65
Ilustración 59: Reportes generado por la librería interna de WebdriverIO en Rundeck.....	68
Ilustración 60: Resultado final de las pruebas en Rundeck generado por la librería interna de WebdriverIO.....	68
Ilustración 61: Pagina de OBE Vacation.....	70
Ilustración 62: Pagina de OBE Room	71
Ilustración 63: Pagina de OBE Flights	71
Ilustración 64: Pagina de OBE Guests.....	72
Ilustración 65: Pagina de OBE Payment.....	72
Ilustración 66: Tablero de tareas para el proyecto de OBE.....	73

ÍNDICE DE TABLAS

Tabla 1: Tecnologías utilizadas en los proyectos	29
Tabla 2: Actividades realizadas Semana 1	34
Tabla 3: Actividades realizadas Semana 2	34
Tabla 4: Actividades realizadas Semana 3	46
Tabla 5: Actividades realizadas Semana 4	46
Tabla 6: Actividades realizadas Semana 5	55
Tabla 7: Actividades realizadas Semana 6	56
Tabla 8: Actividades realizadas Semana 7	56
Tabla 9: Actividades realizadas Semana 8	57
Tabla 10: Actividades realizadas Semana 9	61
Tabla 11: Actividades realizadas Semana 10	61
Tabla 12: Actividades realizadas Semana 11	61
Tabla 13: Actividades realizadas Semana 12	62
Tabla 14: Actividades realizadas Semana 13	62
Tabla 15: Actividades realizadas Semana 14	66
Tabla 16: Actividades realizadas Semana 15	67
Tabla 17: Actividades realizadas Semana 16	67
Tabla 18: Actividades realizadas Semana 17	74
Tabla 19: Actividades realizadas Semana 18	75
Tabla 20: Actividades realizadas Semana 19	75
Tabla 21: Actividades realizadas Semana 20	76
Tabla 22: Actividades realizadas Semana 21	76

Tabla 23: Actividades realizadas Semana 22 77

Tabla 24: Actividades realizadas Semana 23 77

I. INTRODUCCIÓN

En la actualidad, la industria del desarrollo de software ha tenido un gran impacto a nivel mundial, dado que la mayor parte de las empresas buscan estar lo más actualizadas posibles, por lo que integran esta área como parte de sus atractivos para superar la competencia, y por ello deciden invertir en el sector de tecnologías de la información. La compañía de San Services S. de R. L se dedica al rubro del desarrollo de software por lo que en el presente documento se describe a detalle el trabajo realizado en dicha empresa durante la práctica profesional.

En el capítulo II, se describe una leve introducción de la información correspondiente a la empresa de San Services S. de R. L; el rubro al cual se dedica, la jerarquía empresarial y luego se detallan los servicios que ofrece a sus clientes en la actualidad.

En el capítulo III, se presentan las teorías que sustentan el desarrollo del proyecto, relacionadas directamente con el rubro de trabajo, las metodologías de trabajo y las tecnologías utilizadas a lo largo del desarrollo de este.

En el capítulo IV, se encuentra la descripción del trabajo realizado, donde se explica los proyectos en los cuales se trabajó durante el periodo de práctica profesional desde diferentes puntos: proceso de inducción, estructura de los proyectos, explicación sobre de las herramientas y tecnologías utilizadas y se concluye con una bitácora del trabajo realizado.

En el capítulo V, se determinan las conclusiones que se encontraron mediante el cumplimiento de los objetivos planteados en el documento.

Y finalmente en los capítulos VI y se describen las recomendaciones que están orientadas a la personas que deseen realizar trabajos basados en este documento o a los que tenga pensado realizar su práctica profesional en la empresa de San Services S. de R. L.

II. GENERALIDADES DE LA EMPRESA

2.1 DESCRIPCIÓN DE LA EMPRESA

San Services S. de R. L es una empresa dedica al rubro del desarrollo de software desde el año 2007, ha trabajado estrechamente con Sandals Resorts International, proporcionando una variedad de servicios de tecnología de Internet efectivos y confiables adaptados a las necesidades únicas y cambiantes de sus clientes. Los servicios que ofrece son servicios calificados, profesionales y creativos en el diseño web, desarrollo de software, marketing electrónico, SEO, redes sociales, soporte de operaciones de red, diseño gráfico y operaciones de centro de atención telefónica.

El enfoque principal de San Services S. de R. L es integrar a la perfección la experiencia con las últimas tendencias de la industria tecnológica para proporcionar soluciones de desarrollo web personalizadas, innovadoras y adaptativas que siempre estén en constante funcionamiento.

2.2 DESCRIPCIÓN DEL DEPARTAMENTO

El departamento de Quality Assurance dentro de San Services S. de R. L es el encargado de verificar el funcionamiento de las sitios web que se desarrollaron o están en desarrollo para Sandals Resorts International.

El departamento es relativamente nuevo ya que solo cuenta con cuatro integrantes, los cuales son el jefe inmediato del departamento el cual es el encargado de determinar que sitios web tendrán pruebas automatizas, el líder del equipo o "team lead" que es el encargado de determinar qué elementos o que secciones de la página web necesitan ser automatizados, y otros dos miembros los cuales son solo desarrolladores.

2.3 OBJETIVOS

2.3.1 Objetivo General

Desarrollar aplicaciones que se encarguen de realizar pruebas de forma automatizada a los proyectos de desarrollo web de Sandals Resorts International para facilitar y agilizar el proceso de verificación de dichos proyectos.

2.3.2 Objetivos Específicos

- Realizar un proceso de capacitación el cual facilite la incorporación al uso de metodologías, tecnologías y herramientas que serán utilizadas a lo largo del desarrollo de los proyectos.
- Desarrollar nuevas funcionalidades y añadir actualizaciones a proyectos desarrollados previamente por el departamento de Quality Assurance.
- Desarrollar un proyecto que verifique otras funcionalidades de los sitios web de Sandals Resort International y que el cual cumpla con los requerimientos solicitados por el departamento de Quality Assurance como ser la estructura definida en los proyectos, manejo de buenas prácticas de código limpio y creación de pruebas escalables.

III. MARCO TEÓRICO

3.1 RUBRO DEL PROYECTO

3.1.1 CONTROL DE CALIDAD (QUALITY ASSURANCE)

El control de calidad es un procedimiento que se encarga de verificar la calidad de un producto mediante estándares definidos por una compañía tal como lo define Kaoru, (1989):

El control de calidad consiste en el desarrollo, diseño, producción, comercialización y prestación del servicio de productos y servicios con una eficacia del coste y una utilidad óptimas, y que los clientes comprarán con satisfacción. Para alcanzar estos fines, todas las partes de una empresa tienen que trabajar juntos. (p. 2)

En el desarrollo web el control de calidad se aplica para garantizar que los usuarios finales obtengan una interfaz funcional y la mejor experiencia de usuario al usar una aplicación web o sitio web. Por lo que examina un sitio web o una aplicación web para tratar de descubrir cualquier falla que se haya pasado por alto durante el diseño y el desarrollo (Studio, 2019).

3.1.1.1 Objetivo del control de calidad en el software

(Sistema de control de calidad de software - EcuRed, s/f) define los objetivos siguientes:

- Incrementar la productividad y satisfacción en los empleados al mejorar la calidad del producto del software.
- Proveer técnicas aplicadas para automatizar el manejo de datos.
- Realizar una planeación eficaz de los sistemas.
- Documentar.
- Validar y controlar formalmente la calidad del trabajo realizado.
- Cumplir con los objetivos de la organización en cuanto a productividad de sus sistemas de cómputo.

Pero para que el control de calidad realice estos objetivos es necesario, definir los parámetros, indicadores o criterios de medición. El software posee determinados índices medibles que son las

bases para la calidad, el control y el perfeccionamiento de la productividad. Una vez seleccionados los índices de calidad, debe establecerse el proceso de control.

3.1.1.2 Pasos para establecer el proceso de control

(*Sistema de control de calidad de software - EcuRed, s/f*) define los siguientes pasos para lograr establecer el proceso de control:

- **Definir el software que va a ser controlado:** clasificación por tipo, esfera de aplicación, complejidad, etc., de acuerdo con los estándares establecidos para el desarrollo del software.
- **Seleccionar una medida que pueda ser aplicada al objeto de control:** Para cada clase de software es necesario definir los indicadores y sus magnitudes.
- **Crear o determinar los métodos de valoración de los indicadores:** métodos manuales como cuestionarios o encuestas estándares para la medición de criterios periciales y herramientas automatizadas para medir los criterios de cálculo.
- **Definir las regulaciones organizativas para realizar el control:** Quiénes participan en el control de la calidad, cuándo se realiza, qué documentos deben ser revisados y elaborados.

La calidad del software debe implementarse a lo largo de todo el ciclo de vida, desde la planificación del producto hasta la fase de producción de este. Para ello (*Sistema de control de calidad de software - EcuRed, s/f*) añade que existen una serie de ayudas, a través de distintas actividades para la implantación del control de calidad en el desarrollo de software son:

- Aplicación de metodología y técnicas de desarrollo
- Reutilización de procesos de revisión formales
- Prueba del software
- Ajustes a los estándares de desarrollo
- Control de cambios, mediciones y recopilación de información
- Gestión de informes sobre el control de calidad

3.2 METODOLOGÍA DE TRABAJO

3.2.1 DESARROLLO AGILE

Es una metodología de gestión de proyectos adaptativa, que permite llevar a cabo, proyectos de desarrollo de software, adaptándose a los cambios y evolucionando en forma conjunta con el software.

Bahit, (2011) menciona que implementar una metodología de gestión, básicamente nos permite organizar mejor un proyecto y obtener mejores resultados del software entregado al cliente, evitando los fracasos durante el proceso.

Y que fracasos se pueden evitar utilizando esta metodología pues según Bahit, (2011) existen tres motivos para el fracaso que ningún de los cuales ningún proyecto está exento:

- El proyecto lleva más tiempo del que se había planificado.
- El proyecto necesita más dinero del que se había pautado.
- Las funcionalidades del software no dan los resultados esperados.

3.2.1.1 Valores Agiles

Bahit, (2011) define los cuatro valores que impulsan la metodología ágil:

Individuos e interacciones sobre procesos y herramientas: las metodologías ágiles, prefieren valorar la idoneidad de cada individuo, depositando en ésta, la confianza necesaria para lograr una buena comunicación, fluida e interactiva entre todos los participantes del proyecto.

Software funcionando sobre documentación extensiva: las metodologías ágiles proponen destinar todo el tiempo al desarrollo de la aplicación, centrando el mismo, en pequeñas partes operativas y funcionales, que, por un lado, otorguen mayor valor al proyecto y por otro, permitan probar software real.

Colaboración con el cliente sobre negociación contractual: las metodologías ágiles, prefieren integrar al cliente al proyecto, para que de este modo se mas fácil la comunicación directa de éste con el equipo de desarrollo, a fin de generar un ambiente de colaboración, donde los cambios que el cliente proponga no signifiquen un riesgo, sino un aporte al valor del software.

Respuesta ante el cambio sobre seguir un plan: las metodologías ágiles, proponen adaptarse a los cambios sugeridos por el cliente, lo que significa, que, a la modificación propuesta, no se sumarán inconvenientes administrativos, sino al contrario, se acelerarán los procesos, actuando en consecuencia de lo que ha sido solicitado.

3.2.1.2 Doce principios Agiles

Bahit, (2011) también considera que para que la aplicación de la metodología ágil tenga éxito se deben seguir ciertos principios:

1. Satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Tener en cuenta que los requisitos cambian, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregar software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores deben trabajar juntos de forma cotidiana durante el desarrollo del proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Por lo tanto, se debe proporcionar un entorno adecuado y brindar apoyo cuando sea necesario, y confianza a los individuos en la ejecución del trabajo.

No deben existir intermediarios. Todo cambio y toda planificación, se debe realizar entre el cliente y el equipo de desarrolladores.

6. El software funcional es la medida principal de progreso. Por lo tanto, todo software que haya sido entregado debe encontrarse en pleno funcionamiento.
7. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida hasta que el proyecto concluya.
8. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
9. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
10. Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
11. A intervalos regulares el equipo debe reflexionar sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

3.2.2 SCRUM

Scrum es una metodología ágil para la gestión de proyectos relacionados con el desarrollo de software. Entrando más a detalle Pete Deemer, Gabrielle Benefield, Craig Larman y Bas Vodde definen Scrum como:

“Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados Sprints. Los Sprints son de duración fija –terminan en una fecha específica, aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multifuncional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos” (Deemer et al., 2009, p. 5).

El objetivo principal de los Sprint según Bahit, (2011), es crear continuidad y regularidad por lo que se determina en seis reuniones que aseguran el cumplimiento de objetivos:

1. Reunión de Planificación de la Entrega
2. Reunión de Planificación del Sprint
3. El Sprint – corazón de Scrum
4. Reunión diaria
5. Reunión de revisión
6. Reunión de Retrospectiva

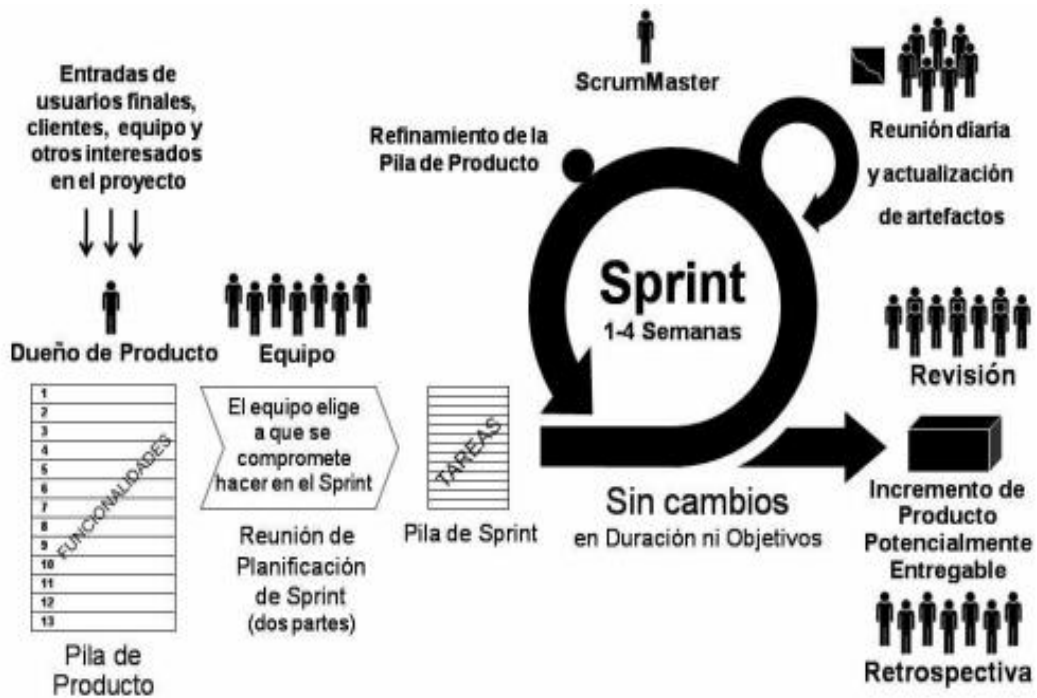


Ilustración 1: Roles, artefactos y eventos principales de Scrum

Fuente: (Deemer et al., 2009)

3.2.2.1 Los roles de SCRUM

Según Bahit, (2011) los equipos de Scrum definen tres roles bien diferenciados:

1. **El Scrum Máster:** Es el alma mater de Scrum. El Scrum máster no es un líder, sino que es un auténtico servidor neutral, que será el encargado de fomentar e instruir sobre los principios ágiles de Scrum.
2. **El dueño de producto:** Es la única persona autorizada para decidir sobre cuáles funcionalidades y características funcionales tendrá el producto. Es quien representa al cliente, usuarios del software y todas aquellas partes interesadas en el producto.
3. **El equipo de desarrollo:** Es el equipo integrado por programadores, diseñadores, arquitectos, testers y demás, que, en forma autoorganizada, será los encargados de desarrollar el producto.

3.2.3 KANBAN

Es una metodología ágil, que cuyo objetivo es gestionar de manera general cómo se van completando las tareas. Kanban es una palabra japonesa que significa "tarjetas visuales", donde Kan es "visual", y Ban corresponde a "tarjeta" (Bahit, 2011).

El sistema Kanban, es representado por tarjetas, que se distribuyen en cantidades limitadas. Cada ítem de trabajo se acompaña de una de estas tarjetas, por lo cual, un nuevo ítem solo podrá iniciarse si se dispone de una tarjeta Kanban libre. Cuando no hay más tarjetas libres, no se pueden iniciar nuevos trabajos. Y cuando un ítem es concluido, una nueva tarjeta se libera, permitiendo el comienzo de un nuevo ítem de trabajo (Bahit, 2011).

3.2.3.1 Tres reglas de Kanban

En el desarrollo de software se capta la esencia de este sistema llevándolo a la práctica mediante la implementación de tres reglas:

- 1. Mostrar el proceso:** Esta regla busca hacer visibles los ítems de trabajo permitiendo conocer de manera explícita el proceso trabajo actual, así como los impedimentos que vayan surgiendo. Dicha visualización, se realiza a través de tableros físicos, al igual que en Scrum, solo que con diferentes columnas.
- 2. Limitar el trabajo en curso:** En Kanban, el límite de trabajo en curso está dado por el WIP: Work in Progress. El límite WIP, consiste en definir la cantidad de ítems simultáneos que pueden ejecutarse en un mismo proceso. Cada proceso, puede tener diferentes límites WIP, como se muestra en la ilustración:

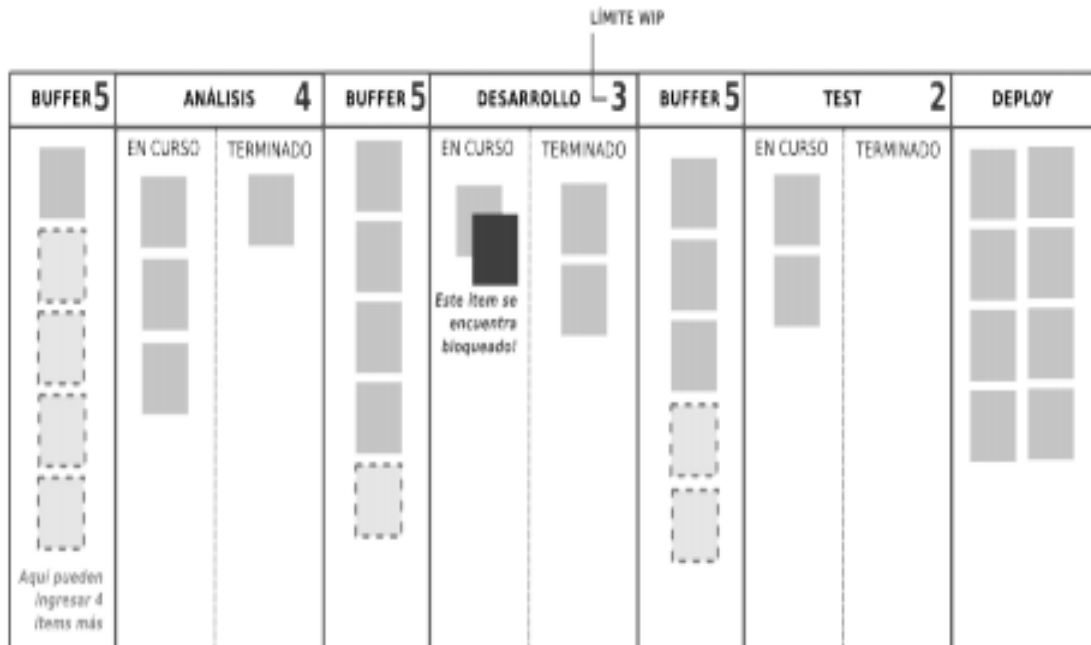


Ilustración 2: Límites WIP.

Fuente: (Bahit, 2011)

Los límites WIP, permiten detectar los cuellos de botella rápidamente. Con solo observar la ilustración prestando atención en las columnas bloqueadas y su procesos anterior y posterior inmediatos, podremos deducir donde se encuentran los puntos de conflicto, y trabajar para remediarlos.

- 3. Optimizar el flujo:** El objetivo de optimizar el flujo de trabajo, es alcanzar un proceso de desarrollo estable, previsible y acorde a las necesidades del proyecto, en el cual se distinga un ritmo de trabajo parejo. Cuando un ítem de trabajo se encuentra estancado, el equipo deberá colaborar a fin de lograr recuperar un flujo parejo y tomar medidas para prevenir futuros estancamientos.

3.3 TECNOLOGÍAS UTILIZADAS

3.3.1 JAVASCRIPT

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

“Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario” (Eguíluz, 2008, p. 5).

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java.

JavaScript hoy en día no solo es utilizado para creación de páginas web dinámicas si no que es utilizado programas de escritorio y servidores siendo NodeJS el más conocido. Así mismos gestores de base de datos como ser el tan popular MongoDB y el más reciente Lovefield de Google.

3.3.1.1 HISTORIA

A principios de los años 90 Brendan Eich, un programador que trabajaba en Netscape pensó en crear un lenguaje de programación que se ejecutara en el navegador del usuario para de este modo evitar los tiempos de carga en aplicaciones web complejas (Eguiluz, 2008).

“Inicialmente, Eich denominó a su lenguaje LiveScript. Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Además, justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript. La razón del cambio de nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época” (Eguiluz, 2008, p. 5).

3.3.1.2 Valores y tipos

Aunque todas las variables de JavaScript se declaran mediante la palabra reservada var, la forma en la que se les asigna un tipo depende del valor que se desea almacenar y entre estos están:

- **Numérico (tipo Number):** Se utilizan para almacenar valores numéricos enteros, decimales y exponenciales.

- **Texto (tipo String):** Se utilizan para almacenar caracteres, palabras y/o frases de texto. Para asignar el valor a la variable, se encierra el valor entre comillas dobles, simples o comillas invertidas, para delimitar su comienzo y su final. De tal manera que "hola", 'hola' y `hola` son considerados valores de tipo string.
- **Booleano (tipo Boolean):** Una variable de tipo boolean almacena un tipo especial de valor que solamente puede tomar dos valores: true (verdadero) o false (falso).
- **Arreglos (tipo Array):** Un array es una colección de variables, que pueden ser todas del mismo tipo o cada una de un tipo diferente.
- **Nulos o sin valor:** se describen como nulos o falta de valor cuando las variables no tienen ningún valor asignado se describen mediante las palabras reservadas: null y undefined.

3.3.1.3 Conversión automática de tipos

JavaScript se podría decir que está escrito a la ligera. Una variable puede contener diferentes tipos de datos, y una variable puede cambiar su tipo de datos a esto se le llama conversión automática de tipos, por lo que 1 es igual a "1" (Simpson, 2015).

```

1 console.log(null + 1); // 1
2 console.log("1" + 2); // 12
3 console.log("2" * "3"); // 6
4 console.log("2" - "3"); // -1
5 console.log(undefined + "Hola"); // undefinedHola

```

Ilustración 3: Conversión automática en JS

Fuente: Propia

3.3.1.4 Funciones y clases

Cuando se desarrolla una aplicación compleja, es muy habitual utilizar una y otra vez las mismas instrucciones. Las funciones son la solución a todos estos problemas, tanto en JavaScript como en el resto de los lenguajes de programación. Eguiluz, (2008) define a una función como un conjunto de instrucciones que se agrupan para realizar una tarea concreta y que se pueden reutilizar fácilmente.

Las clases de JavaScript son una mejora sintáctica sobre la herencia basada en prototipos de JavaScript. (*JavaScript | MDN, s/f*) define a una clase como funciones especiales que no introduce un nuevo modelo de herencia orientada a objetos, pero lo que sí aporta es una sintaxis mucho más clara y simple para crear objetos y lidiar con la herencia.

3.3.2 NODEJS

Node.js es un entorno JavaScript que nos permite ejecutar en el servidor, de manera asíncrona, con una arquitectura orientada a eventos y basado en el motor V8 de Google.

El motor V8 compila JavaScript en código máquina nativo en vez de interpretarlo en el navegador, consiguiendo así una velocidad mucho más alta. NodeJS es de código abierto y puede ejecutarse en varios sistemas operativos como Mac OS X, Windows y Linux.

Además de la alta velocidad de ejecución, Node.js dispone del Bucle de Eventos, que permitirá gestionar enormes cantidades de clientes de forma asíncrona (*Node.js, s/f*).

3.3.3 TRELLO

Es una herramienta para la organización de tareas. Es ideal para la coordinación de equipos de trabajo y se basa en la metodología Kanban, la cual propone un sistema de uso colaborativo, mediante la incorporación de tableros y tarjetas en un espacio de trabajo físico, para coordinar diferentes actividades.

Respecto a su funcionamiento, la distribución de pendientes se realiza en base a un tablero principal, en donde se generan diversas listas con títulos para cada etapa o ítem del proyecto.

3.3.4 BABEL

Babel es una toolchain (cadena de herramientas) que se utiliza principalmente para convertir el código ECMAScript 2015+ en una versión de JavaScript compatible con versiones anteriores en navegadores o entornos actuales y anteriores (*Usage Guide · Babel, s/f*).

3.3.5 WEBDRIVERIO

WebDriverIO es una herramienta creada en JavaScript que permite controlar un navegador o una aplicación móvil con solo unas pocas líneas de código. (*Getting Started · WebdriverIO, s/f*). En pocas palabras funciona como un enlace para el protocolo WebDriver además de que es un marco de pruebas completo con numerosas características y utilidades adicionales lo que lo convierte en una herramienta muy utilizada para realizar pruebas en aplicaciones web o móviles de forma automatizada y muy sencilla.

(Red Hat, s/f) define a la automatización en el área de la informática como el uso de sistemas de software para crear instrucciones y procesos repetibles a fin de reemplazar o reducir la interacción humana con los sistemas de TI. El software de automatización funciona dentro de los límites de esas instrucciones, herramientas y marcos para realizar las tareas con muy poca intervención humana.

WebDriverIO también mejora el paquete WebDriver con comandos y opciones adicionales disponibles que se pueden configurar en el archivo de configuración del WDIO testrunner.

3.3.5.1 WebDriver

La API WebDriver se define mediante un conjunto de interfaces para descubrir y manipular elementos del DOM en una página web, y para controlar el comportamiento del navegador que lo contiene.

WebDriver dispone de las siguientes opciones predeterminadas:

- **Protocol (Protocolo):** Protocolo a utilizar al comunicarse con el servidor del controlador.
- **Hostname:** Host de su servidor de controladores.
- **Port (Puerto):** Puerto donde se encuentra su servidor.
- **Path:** Dirección de donde se encuentra el controlador del servidor.
- **QueryParameters:** Parámetros que se propagan al controlador del servidor.
- **Capabilities:** Define las capacidades que desea ejecutar en su sesión de WebDriver.
- **LogLevel:** Nivel de verbosidad de registro. Entre estas opciones están trace, debug, info, warn, error y silent.

- **OutputDir:** Directorio para almacenar todos los archivos de registro de testrunner (incluidos los registros generados por reportes y los registros de wdio). Si no se establece un directorio todos los registros serán mostrados en la consola.
- **ConnectionRetryTimeout:** Tiempo de espera para cualquier solicitud de WebDriver a un controlador o una red.
- **ConnectionRetryCount:** Recuento máximo de reintentos de solicitud al servidor.
- **Agent:** Le permite utilizar un agente personalizado http o https para realizar solicitudes al servidor.
- **Headers:** Especifica encabezados personalizados para pasar a cada solicitud.

3.3.5.2 WDIO testrunner

El WDIO testrunner agrega funciones adicionales como ser:

- **Specs:** Define cuales son los archivos de prueba que serán ejecutados.

```
specs: [
  'test/spec/**'
],
```

Ilustración 4: Configuración de specs

Fuente: Propia

- **Exclude:** Define que pruebas serán excluidas de la ejecución.

```
exclude: [
  'test/spec/multibrowser/**',
  'test/spec/mobile/**'
],
```

Ilustración 5: Configuración de exclude

Fuente: Propia

- **Suites:** Es un objeto que describe varias suites, que luego puede especificar con la opción --suite en la CLI de wdio.

- **Capabilities:** Es igual que la sección de capacidades descrita anteriormente, excepto que da la opción de especificar un objeto de múltiples sesiones de WebDriver en una matriz para realizar ejecución paralela.

```
capabilities: [{
  maxInstances: 5,
  browserName: 'firefox',
  specs: [
    'test/ffOnly/*'
  ],
}],
```

Ilustración 6: Configuración de capabilities

Fuente: Propia

- **BaseUrl:** Define cual será la primera dirección URL a la cual acceden las pruebas si no se define ninguna.
- **Bail:** Define la tolerancia a errores en las pruebas.
- **SpecFileRetries:** Define cuantas veces se repetirá una prueba en caso de que este falle.
- **WaitforTimeout:** Define el tiempo de espera máximo para los comandos waitFor* en las pruebas.
- **WaitforInterval:** Intervalo predeterminado para todos los comandos waitFor* para verificar si se ha cambiado un estado esperado (por ejemplo, visibilidad).
- **Services:** Define que WebDriver se utilizaran en la ejecución de las pruebas, se pueden utilizar varios WebDrivers por ejemplo ChromeDriver para el navegador Chrome y GeckoDriver para Mozilla Firefox.
- **Framework:** Define el framework que utilizará el WDIO testrunner entre estos tenemos Mocha, Cucumber y Jasmine.

```
services: ['selenium-standalone', 'chromedriver', 'geckodriver'],
framework: 'mocha',
```

Ilustración 7: Configuración de services y framework

Fuente: Propia

- **MochaOpts:** Define cuales son las opciones que puede realizar el framework Mocha.

- **CucumberOpts:** Define cuales son las opciones que puede realizar el framework Cucumber.
- **JasmineNodeOpts:** Define son las opciones que puede realizar el framework Jasmine.
- **Reporters:** Define la herramienta que generara los reportes de las pruebas. Pueden ser más de uno.

```
reporters: [
  'dot',
  ['allure', {
    outputDir: './'
  }]
],
```

Ilustración 8: Configuración de reporters

Fuente: Propia

- **AutomationProtocol:** Define el protocolo que se usara para realizar la automatización en un navegador. Actualmente solo se admiten WebDriver y devtools.
- **User:** El nombre de usuario que da acceso a su servidor en la nube.
- **Key:** El clave de acceso o clave secreta de su servidor en la nube.
- **Región:** define la región donde el servidor ejecutara las pruebas.

```
user: 'webdriverio',
key: 'xxxxxxxxxxxxxxxxxxxx-xxxxxx-xxxxxx-xxxxxxxxxx',
region: 'us',
```

Ilustración 9: Configuración de user, key y región

Fuente: Propia

- **Hooks:** permite establecer el uso de hooks (ganchos) para que se activen en momentos específicos del ciclo de vida de la prueba. Esto permite acciones personalizadas, por ejemplo, tomar una captura de pantalla si falla una prueba como se muestra en la siguiente ilustración.

```
afterStep: function (uri, feature, { error, result, duration, passed }, stepData, context) {  
  if(error != undefined){  
    browser.takeScreenshot();  
  }  
},
```

Ilustración 10: Ejemplo de un hook

Fuente: Propia

3.3.5.3 Estructura

WebDriverIO fue diseñado tomando en cuenta el Page Object Pattern (patrón de objetos de página) que se basa en crear una clase específica para cada página web, lo que permitió construir grandes conjuntos de pruebas utilizando este patrón (*Getting Started · WebdriverIO, s/f*).

“Resulta que las clases limpias y modernas proporcionan todas las características necesarias como ser la herencia entre objetos de página, carga perezosa de elementos, encapsulación de métodos y acciones” (*Getting Started · WebdriverIO, s/f*).

Esta estructura de colocar todos los captadores (getters) y funciones en un Objeto de página, puede importar dentro de las especificaciones y mantener su código legible. Esto significa que sus especificaciones sirven como documentación para personas no técnicas. Además, si falla un bloque “it” (en caso de estar usando Mocha), al seguir esta estructura de codificación, es más fácil para los desarrolladores web depurar los errores.

```
2  export default class Page {  
3    constructor() {  
4      this.title = 'My Page'  
5    }  
6  
7    open(path) {  
8      browser.url(path)  
9    }  
10 }  
11
```

Ilustración 11: Estructura básica de un page object

Fuente: Propia

3.3.6 MOCHA JS

Mocha es un framework de pruebas de JavaScript que se ejecuta en Node.js. Nos da la posibilidad de crear tanto pruebas síncronas como asíncronas de una forma muy sencilla. Además de proporcionar muchas utilidades para la ejecución y el reporte de las pruebas como por ejemplo Mochawesome (*Mocha - the fun, simple, flexible JavaScript test framework, s/f*).

3.3.6.1 Estructura

Las pruebas creadas en Mocha siguen una estructura determinada por describe (descripción) y it:

- **Describe:** Utilizamos este elemento para definir bloques de pruebas relacionadas entre sí. Podemos concatenar varios "describe" si lo vemos necesario para nuestra estructura de pruebas.
- **It:** Cada elemento "it" sería una prueba. Podemos definir todas las pruebas (it) que sean necesarias dentro de un elemento describe. Las pruebas pueden estar divididas entre varios elementos de tipo "describe". Este es útil para diferenciar varios bloques de pruebas en un mismo fichero.

```
1 describe("Deberia realizar una operacion matematica", function() {
2   |   it("Revisar si la suma de A y B da como resultado 4", function() {
3     |     expect(A+B).to.be.equal(4);
4   |   });
5 });
```

Ilustración 12: Ejemplo del uso de it y describe

Fuente: Propia

- **Hooks:** Se colocan dentro de un elemento de tipo describe y son utilizados mayormente para acciones tan simples y necesarias como ser inicializar una variable antes de la ejecución de un it específico o limpiar una base de datos después de que una prueba fallara.
- **Exclusive Tests:** La función only hace posible ejecutar solo el bloque o el caso de prueba al cual se lo añadamos. Es muy útil si tenemos un gran número de pruebas y por algún motivo queremos ejecutar un parte de ellos o incluso una sola prueba.

- **Inclusive Tests:** Podemos decir que la función skip se comporta de forma opuesta a only. Con skip podemos provocar que, durante una ejecución, no se ejecute un bloque o incluso una prueba dentro del mismo.

3.3.6.2 Mochawesome

Mochawesome es un generador de reportes personalizado para usar con el marco de prueba de JavaScript y Mocha. Se ejecuta en Node.js y funciona junto con la librería de mochawesome-report-generator para generar un informe HTML / CSS independiente para ayudar a visualizar los resultados finales de las ejecuciones de pruebas.

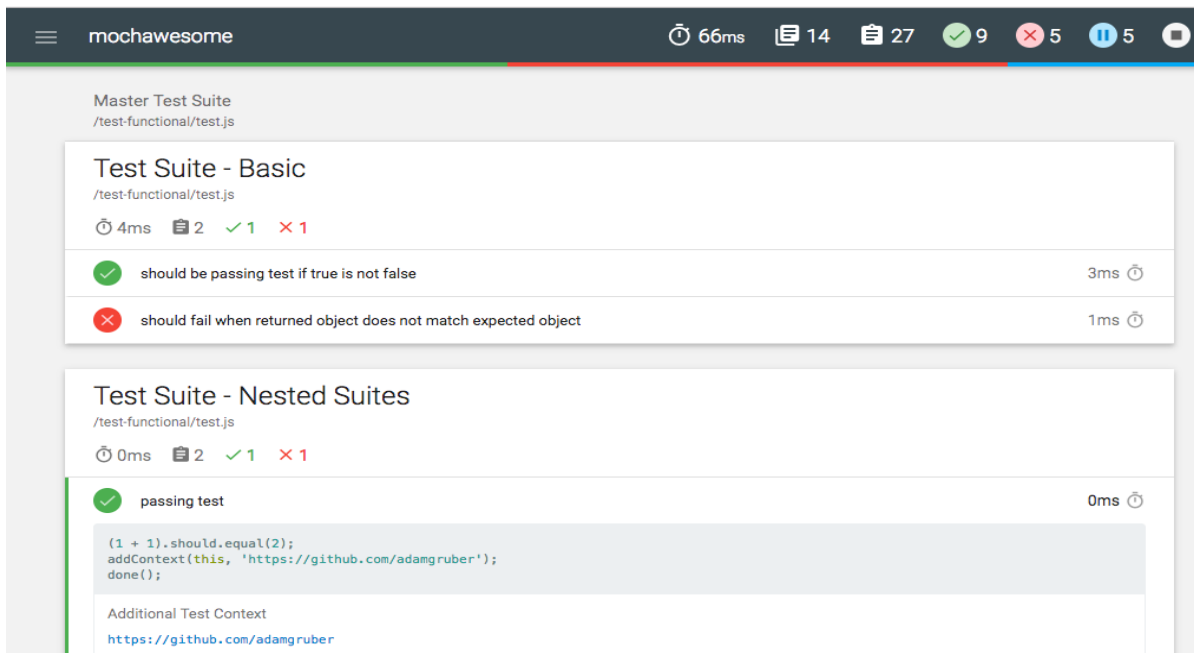


Ilustración 13: Reporte generado por Mochawesome

Fuente: (adamgruber/mochawesome: A Gorgeous HTML/CSS Reporter for Mocha.js, s/f)

3.3.7 CHAI JS

Es una librería de aserciones, la cual se puede emparejar con cualquier marco de pruebas de JavaScript. Chai tiene varias interfaces: assert, expect y should, que permiten al desarrollador elegir el estilo que le resulte más legible y cómodo a la hora de desarrollar sus pruebas (*Expect / Should - Chai, s/f*).

3.3.7.1 Interfaces

(*Expect / Should - Chai, s/f*) da una descripción de cada una de las interfaces:

- **Expect:** La interfaz expect funciona para verificar que el resultado de una operación o un método sea el esperado, además expect permite incluir mensajes arbitrarios para anteponer cualquier afirmación fallida que pueda ocurrir.

```
1 expect(2).to.equal(2); // Verdadero
2 expect(2).to.equal(1); // Falso
```

Ilustración 14: Ejemplo de expect

Fuente: Propia

- **Should:** El estilo should permite las mismas aserciones encadénales que la interfaz expect, sin embargo, extiende cada objeto con una propiedad should para iniciar su cadena.

```
1 let foo = 'bar';
2 foo.should.be.a('string'); // Verdadero
3 foo.should.equal('bar'); // Verdadero
4 foo.should.have.lengthOf(1); // Falso
```

Ilustración 15: Ejemplo de should

Fuente: Propia

3.3.8 SELENIOD

Selenoid es una herramienta de código abierto que ayuda a ejecutar distintos navegadores en contenedores de Docker. Cabe añadir que se crea un nuevo contenedor para cada nueva solicitud del navegador y lo detiene una vez este finalice su sesión (*Selenoid UI, s/f*).

Cada contenedor de Selenoid tiene una versión determinada del navegador, la versión requerida del controlador web o el servidor Selenium, que admite el navegador y todas las dependencias necesarias. Además, todos los procesos del navegador están completamente aislados. Esto permite iniciar simultáneamente un número ilimitado de múltiples versiones de navegador.

Selenoid cuenta su propia interfaz de grafica a la cual se puede acceder usando nuestro localhost en un navegador, en esta interfaz se pueden observar las sesiones que están siendo ejecutadas con su respectiva información.

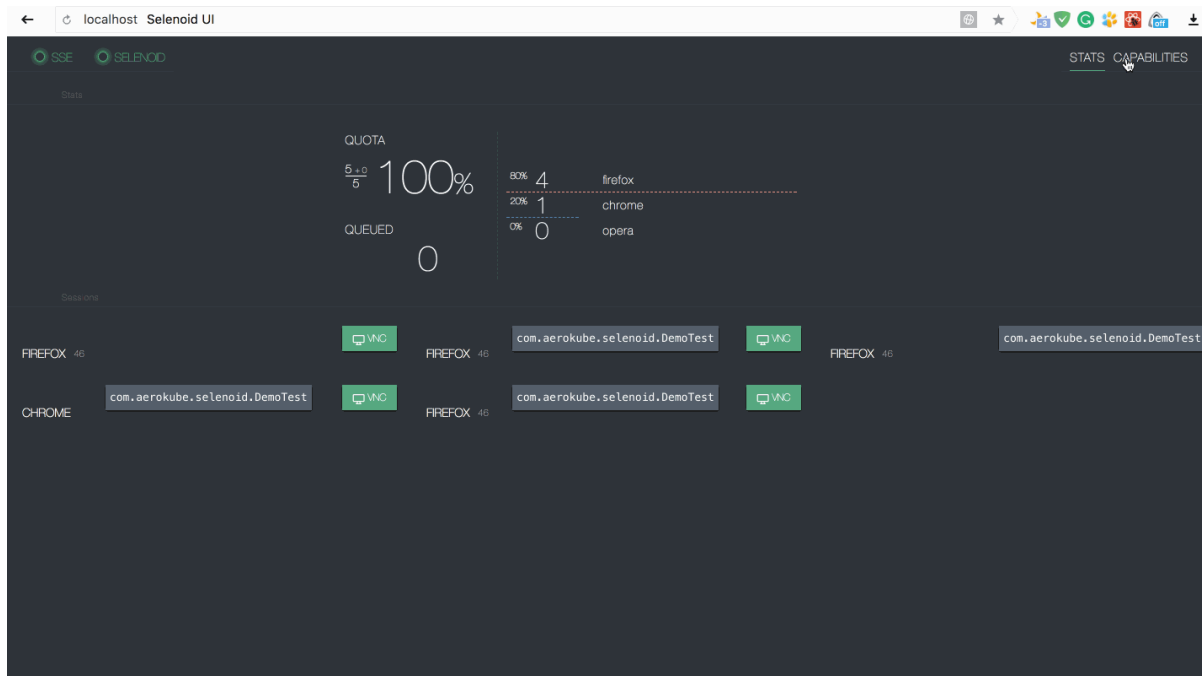


Ilustración 16: Interfaz gráfica de Selenoid

Fuente: (*Selenoid UI, s/f*)

3.3.8.1 Historia

Selenoid fue creado en octubre del 2017 por Aerokube esta herramienta fue producida con lenguaje de programación Golang y fue desarrollado en base al del código original de Selenium Hub. (*Selenoid UI, s/f*). Hoy en día Selenoid es compatible con todos los navegadores web populares.

3.3.9 GIT

Git es un sistema de control de versiones distribuido de forma gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta muy grandes, con velocidad y eficiencia.

Chacon & Straub, (2009) define que un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que se puedan recuperar versiones específicas en un futuro.

3.3.9.1 Historia

Git surgió en 2005 como una necesidad de Linus Torvalds creador de Linux ya que durante la mayor parte del mantenimiento del kernel de Linux (1991-2002), los cambios en el software se realizaban a través de parches y archivos. Fue hasta el año 2002, en que el proyecto del kernel de Linux empezó a usar un DVCS propietario llamado BitKeeper. Pero en el año 2005, la relación entre la comunidad que desarrollaba el kernel de Linux y la compañía que desarrollaba BitKeeper se vino abajo y la herramienta dejó de ser ofrecida de manera gratuita. Esto impulsó a la comunidad de desarrollo de Linux a desarrollar su propia herramienta basada en algunas de las lecciones que aprendieron mientras usaban BitKeeper (Chacon & Straub, 2009).

Desde entonces Git ha evolucionado y madurado para ser fácil de usar y conservar sus características iniciales.

3.3.9.2 Fundamentos

Copias instantáneas de archivos, no diferenciales: Cada vez que se realiza un cambio en un proyecto, o se guarda el estado actual del proyecto, Git básicamente toma una foto del aspecto de todos tus archivos en ese momento y guarda una referencia única a esa copia instantánea (Chacon & Straub, 2009).

Casi todas las operaciones son locales: La mayor parte de las operaciones que se pueden realizar en Git sólo necesitan archivos y recursos locales para funcionar. En pocas palabras no se necesita información de ningún otro computador que se encuentre en tu red (Chacon & Straub, 2009).

Integridad: Todo es verificado mediante una suma de comprobación antes de almacenar algún cambio, y es identificado a partir de ese momento mediante dicha suma. Lo que da a entender que es imposible cambiar los contenidos de cualquier archivo o directorio sin que Git lo note (Chacon & Straub, 2009).

Solo añade información: Cada acción realizada en Git solamente añade información por lo que en ningún momento se borra información de un proyecto, por lo que es muy difícil que el sistema realice una acción que no se enmendar (Chacon & Straub, 2009).

Los tres estados: Git tiene tres estados principales en los que se pueden encontrar tus archivos:

- Confirmado: significa que los datos están almacenados de manera segura en tu base de datos local.
- Modificado: significa que has modificado el archivo, pero todavía no lo has confirmado a tu base de datos.
- Preparado: significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.

Esto tres estados nos llevan a las tres secciones principales que conforman un proyecto de Git: El directorio de Git (Git directory), el directorio de trabajo (working directory), y el área de preparación (staging área) (Chacon & Straub, 2009).

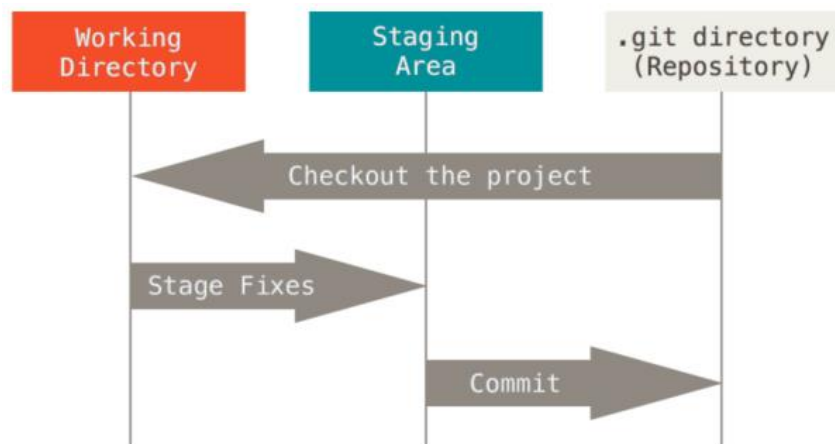


Ilustración 17: Directorio de trabajo, área de preparación y el directorio Git.

Fuente: (Chacon & Straub, 2009)

3.3.9.3 Operaciones Básicas

Inicializar un repositorio: Esto es usado para crear un subdirectorio nuevo llamado git, el cual contiene todos los archivos necesarios del repositorio.

```
$ git init
Initialized empty Git repository in C:/Users/anton/Documents/testing/.git/
```

Ilustración 18: Inicializando un nuevo repositorio

Fuente: Propia

Dar seguimiento a un archivo nuevo: Esto es usado para dar seguimiento a las modificaciones realizadas en un archivo o para agregar ese cambio al repositorio.

```
$ git add helloworld.js|
```

Ilustración 19: Seguimiento de archivos

Fuente: Propia

Confirmar cambios: Esto es usado para registrar una instantánea en el repositorio y confirmar un cambio añadiendo un mensaje dando una leve descripción de este.

```
$ git commit -m 'agregando el archivo helloworld'|
```

Ilustración 20: Confirmando cambios

Fuente: Propia

Subir cambios a un repositorio remoto: Esto es usado para agregar las instantáneas almacenadas localmente en el subdirectorio git y añadirlas a un repositorio remoto.

```
$ git push origin master|
```

Ilustración 21: Subiendo cambios al repositorio remoto

Fuente: Propia

Descargar cambios de un repositorio remoto: Esto es usado para agregar todas las instantáneas de un repositorio remoto a uno local.

```
git pull origin master|
```

Ilustración 22: Descargando cambios de un repositorio remoto

Fuente: Propia

Clonar un repositorio: Esto es usado para descargar un repositorio que se encuentra en un servidor a nuestro directorio local.

```
$ git clone https://github.com/Antonio21MP/weather-app
```

Ilustración 23: Clonando un repositorio

Fuente: Propia

3.3.10 RUNDECK

Rundeck es un software de código abierto que se utiliza para automatizar procedimientos operativos rutinarios en centros de datos o entornos de la nube. Rundeck comenzó en 2010 como un proyecto de código abierto creado por Damon Edwards, Alex Honor y Greg Scheler (*Rundeck Documentation, s/f*).

Rundeck dispone de una API web que permite ejecutar y administrar trabajos, actualizar los recursos de algún nodo y recuperar el historial de ejecución. Además, que se puede utilizar en con otras herramientas, scripts y servicios.

The screenshot shows the Rundeck web interface. At the top, there is a navigation bar with 'RUNDECK' and menu items like 'anvils', 'Jobs', 'Nodes', 'Commands', and 'Activity'. Below the navigation bar, the main content area displays a job execution summary for '#3 Restart anvil/web'. The job is marked as 'Succeeded' and completed after 9 seconds at 5:58 pm. A 'Node Summary' table shows 100% completion (2/2 nodes). Below this, a table lists the nodes and their execution steps, all of which are 'OK'. The 'Activity for this Job' section at the bottom shows filters for 'running', 'recent', 'failed', and 'by you'.

Node	Start time	Duration
www1.anvils.com		0.00:03
anvil/web/stop	5:58:45 pm	0.00:02
anvil/web/start	5:58:48 pm	0.00:01
www2.anvils.com		0.00:03
anvil/web/stop	5:58:47 pm	0.00:01
anvil/web/start	5:58:49 pm	0.00:02

Ilustración 24: Interfaz visual de Rundeck

Fuente: (*Rundeck Documentation, s/f*)

IV. DESCRIPCIÓN DEL TRABAJO REALIZADO

El rubro de San Services S. de R. L es el desarrollo web o web development (en inglés), cuenta con varios equipos de que se encargan del desarrollo web como ser equipos de Front-End, Back-End, Software Quality Assurance(QA) entre otros, por lo que la empresa me asigno formar parte del equipo de Software Quality Assurance en el cual se me fue asignado a distintos proyectos, unos ya existentes y a unos nuevos a medida fueron requeridos. Al comienzo como a cualquier nuevo integrante de un equipo fui asignado a un proceso de inducción y capacitación de manera que yo pudiera ir conociendo como trabajan los desarrolladores en el equipo al que fui asignado, las metodologías que estos utilizan y la estructura general de los proyectos para asegurar una adaptación efectiva al equipo de desarrollo. El tiempo invertido como desarrollador QA en la empresa San Services S. de R. L fue de 26 semanas, comenzado el 9 de octubre del año 2019 y finalizando el 1 de abril del año 2020.

La descripción del trabajo realizado en San Services S. de R. L se dividió en dos partes: el proceso de capacitación y los proyectos como miembro del equipo de desarrolladores de QA.

4.1 PROCESO DE CAPACITACIÓN

En esta sección se definirá el proceso de capacitación, en el cual se explica detalladamente el proceso de inducción, capacitación e integración al esquema de trabajo y tecnologías utilizadas por el equipo de desarrollo de QA.

4.1.1 Tecnologías utilizadas en los proyectos

Previo a la introducción de los proyectos, se llevó a cabo una pequeña charla sobre las tecnologías que se utilizarían en todos los proyectos las cuales se detallan en la siguiente tabla.

Tabla 1: Tecnologías utilizadas en los proyectos

Tecnología	Descripción
JavaScript	Lenguaje de programación utilizado para el desarrollo de los proyectos.
NodeJS	Herramienta para compilar aplicaciones en JavaScript.
WebDriverIO	Herramienta que permite controlar un navegador.
Selenium	Herramienta que ayuda a ejecutar distintos navegadores en contenedores de Docker.
Rundeck	Software que se utiliza para automatizar procedimientos operativos rutinarios.
Yarn	Administrador de paquetes.
Beanstalk	Herramienta utilizada para el manejo de los repositorios.
Mocha	Herramienta utilizada para crear test(pruebas)
Trello	Herramienta utilizada para organizar el trabajo a realizar para cada integrante del equipo.
Slack	Herramienta utilizada para que los miembros del equipo pueden comunicarse y trabajar juntos.

Fuente: Propia

De todas las tecnologías antes mencionadas en la que se llevó mayor tiempo fue con WebDriverIO y Mocha. WebDriverIO es la encargada de manipular las páginas web y cabe añadir que brinda con una amplia documentación que invita a sus usuarios a seguir una estructura sugerida para realizar las pruebas automatizadas de manera apropiada al utilizarlo junto a Mocha.

Cabe añadir que se inculco que los test(pruebas) hechos con Mocha deben seguir un el estilo BDD (Behavior Driven Development, o sea, desarrollo dirigido por comportamiento en español) que es una estrategia de desarrollo parecida a TDD (Test Driven Development o desarrollo dirigido por pruebas) ya que BDD se enfoca en la prueba de más alto nivel, la prueba funcional, la de aceptación, el foco está en cumplir con el negocio y no solo con el código. Otra de las razones por las cuales se inculco el estilo BDD es porque esta encaja bien en las metodologías ágiles, ya que generalmente en ellas se especifican los requerimientos como historias de usuario.

Otra de las áreas en la cuales se llevó un mayor impacto fue en la obtención de selectores del DOM para poder asignárselo a las funciones de WebDriverIO ya que como mencione anteriormente la mayoría de sus funciones depende de que se le provea un selector para poder realizar las acciones en el navegador y para finalizar se realizó una pequeña explicación para el uso de Trello debido a que con dicha herramienta se le asigna a cada integrante del equipo una tarea en específico.

4.1.1.1 Estructura general de los proyectos

Los proyectos siguen una estructura general llamada Page Object Pattern (patrón de objetos de página) que se basa en crear una clase específica para cada página web, esta estructura es recomendada por WebDriverIO ya que la herramienta gira en base a esta estructura. El proyecto amplió un poco más este concepto a modo que no se limita crear una clase específica para cada página web si no que ahora se expande en el hecho de crear clases para secciones globales como ser la región, el sitio principal al que pertenecen dichas páginas web y a secciones específicas de las páginas web como ser botones, imágenes, etc. En la ilustración 25 se muestra dicha estructura.

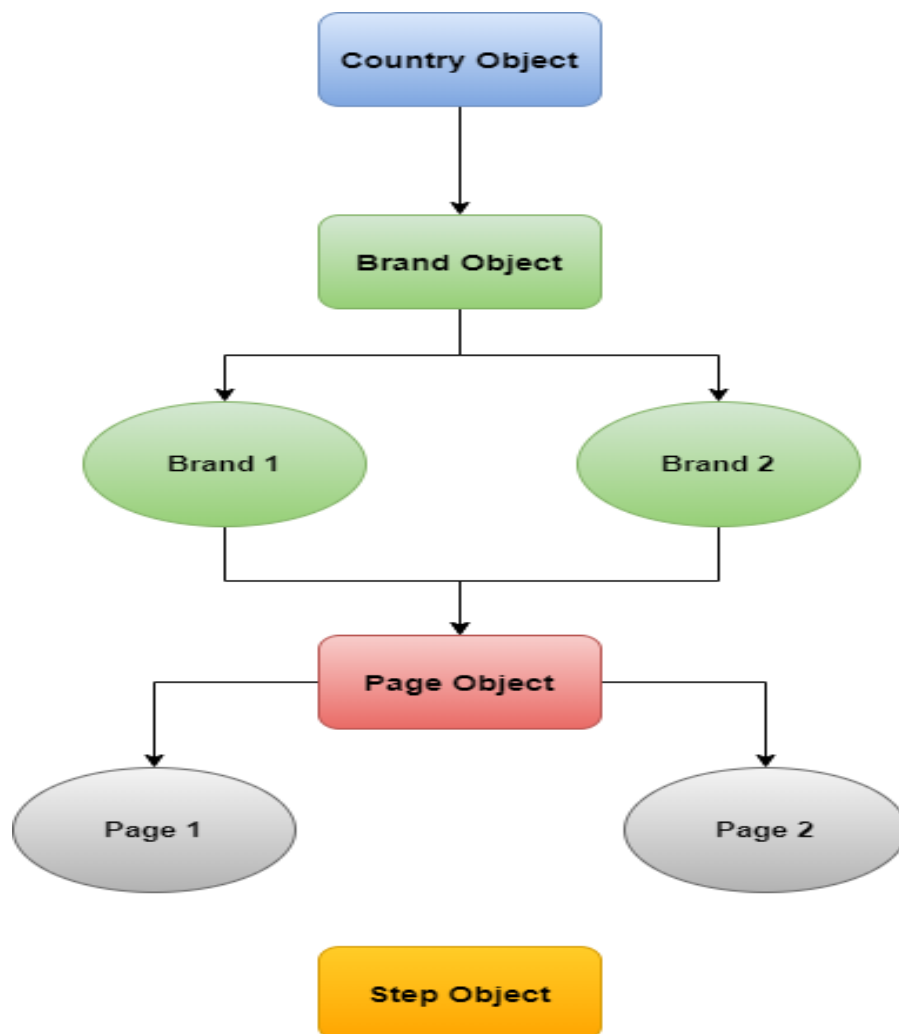


Ilustración 25: Estructura de los proyectos

Fuente: Propia

En la ilustración se puede observar que el proyecto esta seccionado en 4 partes importantes:

Country Object: Son clases que contienen todas las funciones, atributos y selectores necesarios según la región del sitio web. Por ejemplo, la página web de Sandals en Estados Unidos con la del Reino Unido no son del todo iguales ya que esta tiene selectores muy diferentes a los que tiene la de Estados Unidos.

Brand Object: Son clases que contienen todas las funciones, atributos y selectores necesarios según el sitio web. Por ejemplo, la página web de Sandals y Beaches tendrían una clase para cada una ya que no tienen la misma apariencia, por lo tanto, sus selectores, botones e imágenes son diferentes.

Page Object: Los page objects son clases creadas principalmente para contener todos los selectores, funciones y atributos necesarios para realizar pruebas a una página web. Cada página web tiene su propia clase.

Step Object: Los step objects son clases que existe para reducir la duplicación de código ya que cada elemento de la página que sea necesario manipular tendrá su propia clase, por ejemplo, las imagines, las ventanas emergente, etc. tendrán su propia clase para que de este modo sea más fácil manipularlas y reducir las líneas de código en cada spec.

Anteriormente se mencionó que el step object son clases que se utilizan para reducir el código en los specs. Pero que son los specs.

Spec: Los specs son básicamente las pruebas para WebDriverIO y son los que harán uso de todas las clases antes mencionadas. La estructura de un spec se muestra en la siguiente ilustración.

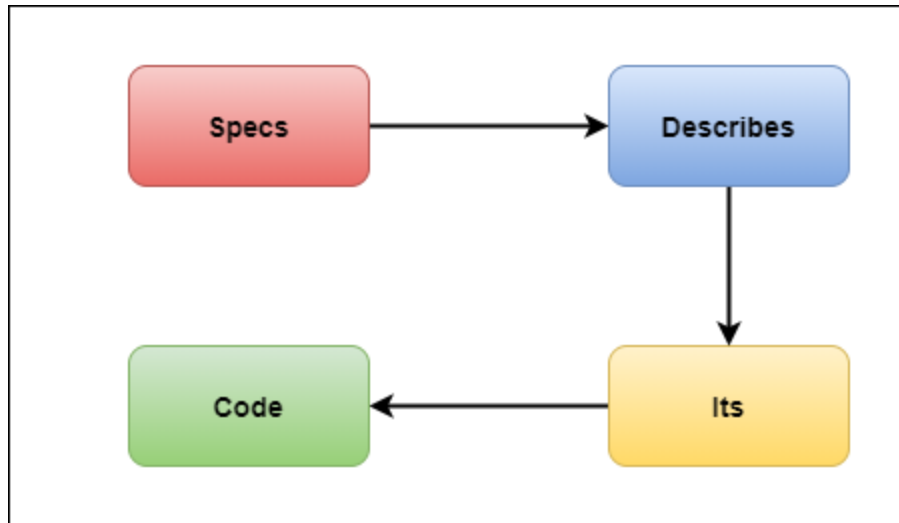


Ilustración 26: Estructura de un spec.

Fuente: Propia

Como se puede ver en la ilustración los specs hacen uso de describes y its que son métodos que pertenecen a Mocha para llevar a cabo la ejecución de las pruebas.

4.1.1.2 Macaw y el archivo de configuración de los proyectos

Además, realizar una introducción a lo que es la estructura del proyecto también se brindó una breve explicación sobre el uso de una herramienta interna del equipo llamada Macaw que básicamente es un tipo de interprete que facilita la manipulación de los specs a gran escala determinando que specs se desea ejecutar, en que host, puerto y región, y al final de su ejecución envía el reporte generado por Mochawesome a un canal de Slack específico. Esta herramienta depende de un archivo de configuración tipo JSON para realizar este funcionamiento. La estructura de este archivo se muestra en la siguiente ilustración:

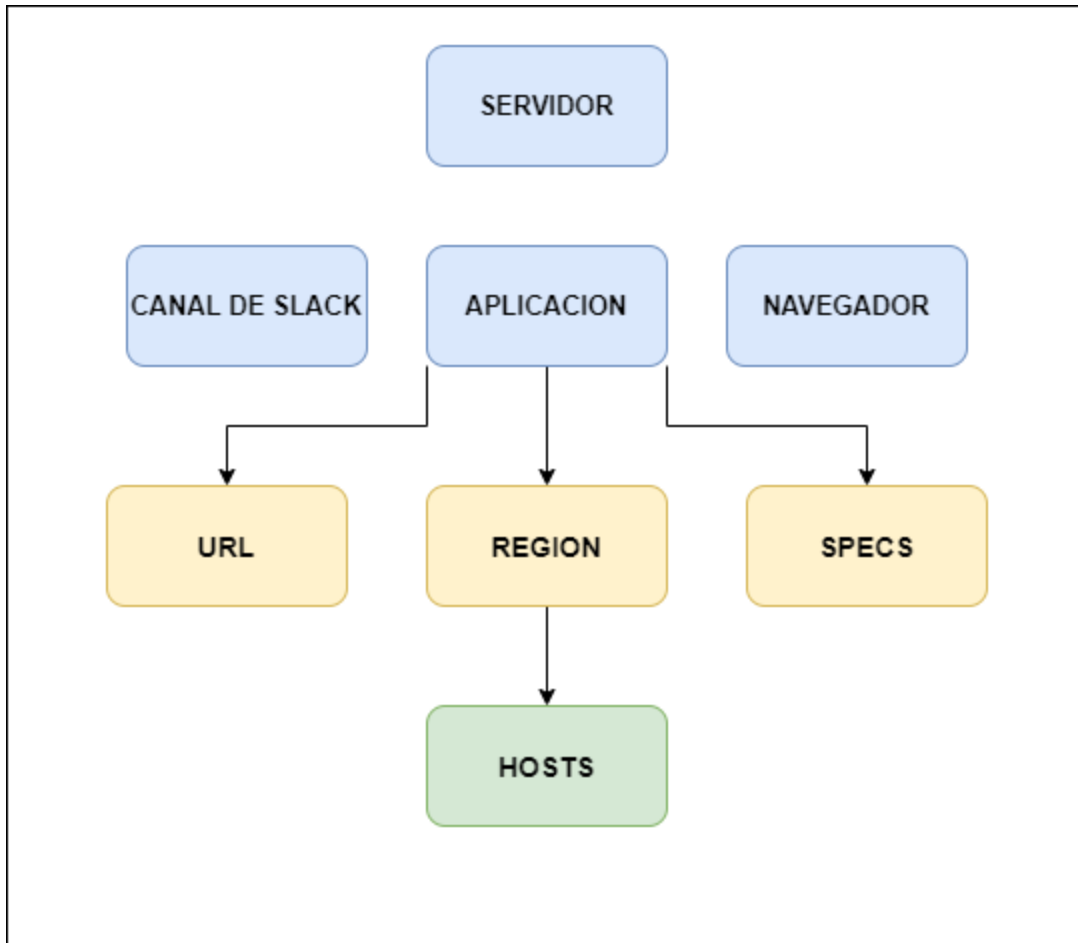


Ilustración 27: Estructura del archivo de configuración

Fuente: Propia

Como se puede ver en la ilustración este archivo está seccionado en cuatro partes que son:

- Servidor en el que se ejecutarán el proyecto.
- Canal de Slack donde se enviará el reporte generado por Mochawesome
- Navegadores en el cual se ejecutarán las pruebas
- Aplicación que se ejecutará. Esta aplicación contiene otras secciones como ser URL del sitio web, la región del servidor con sus respectivos host y los specs que tendrá esta aplicación.

4.1.1.3 Bitácora de actividades realizadas

Tabla 2: Actividades realizadas Semana 1

Fecha	Descripción
Miércoles 9 de octubre del 2019	Se hicieron todas configuraciones correspondientes para poder ser parte del equipo de QA. Asignación de correo electrónico, adición a los grupos de Skype, adición a los grupos de Slack, compartir credenciales de las herramientas que utilizan en los desarrollos.
Jueves 10 de octubre del 2019	Inducción a las tecnologías que se utilizan en la creación de proyectos de QA. Como ser NodeJS, WebDriverIO, Rundeck entre otros.
Viernes 11 de octubre del 2019	Se realizaron lecturas de la documentación de WebDriverIO y Mocha.

Fuente: Propia

Tabla 3: Actividades realizadas Semana 2

Fecha	Descripción
Lunes 14 de octubre del 2019	Hice "Pair Programming" con un miembro del equipo para poder observar y aprender la estructura de los proyectos.
Martes 15 de octubre del 2019	Hice "Pair Programming" con un miembro del equipo para poder observar y aprender la estructura que deben tener los test en Mocha.
Miércoles 16 de octubre del 2019	Hice "Pair Programming" con un miembro del equipo para poder observar y aprender sobre el uso WebDriverIO testrunner.
Jueves 17 de octubre del 2019	Hice "Pair Programming" con un miembro del equipo para poder observar y aprender el uso de las funciones que provee WebDriverIO y la obtención de selectores para estas.
Viernes 18 de octubre del 2019	Hice "Pair Programming" con un miembro del equipo para poder observar y aprender el uso de Macaw y el archivo de configuración.

Fuente: Propia

Como se muestra en las tablas 3 y 4 el proceso de capacitación solo se basó en aprender las tecnologías necesarias para manipular los proyectos, la curva de aprendizaje en estas primeras semanas fue bastante elevada ya que el aprendizaje de herramientas para realizar pruebas era un área muy poco explorada en la carrera. Además, fue muy benéfico el hecho de realizar "Pair Programming" ya que las dudas se aclaraban al instante.

4.2 PROYECTOS COMO PARTE DEL EQUIPO DE DESARROLLO

En esta sección se detallará los proyectos asignados como parte del equipo de desarrollo de QA, explicando a detalle cada uno de los proyectos y el trabajo asignado durante el periodo de

práctica, además se explicará cuáles fueron las razones o circunstancias por las cuales surgieron dichos proyectos y como estos aportan al momento de solucionar dichos problemas.

4.2.1 QA Contents

El proyecto de QA Contents es el proyecto más nuevo del equipo de desarrollo y el cual su total desarrollo me fue asignado. El tiempo que se estuvo laborando en el proyecto fue aproximadamente de 14 semanas, comenzando su desarrollo el lunes 21 de octubre del 2019 y finalizando viernes 24 de enero del 2020. QA Contents fue el proyecto en el cual estuve laborando más tiempo. Cabe añadir que a pesar de ser un proyecto el cual todo su desarrollo dependió de mi persona, el Lead del equipo de QA realizaba reuniones conmigo y demás restos del equipo para determinar que sería testeado de forma automatizada y que no.

4.2.1.1 Definición del Problema

El equipo de desarrollo tiene dos proyectos que testean ciertas secciones de las páginas web de Sandals y Beaches pero no cuentan con un proyecto que abarque todas las paginas principales del sitio por ende hay ciertos sitios los cuales no son testeados de manera automatizada y solo son revisados de forma manual por el equipo de QA Manual de India, por lo el equipo de desarrollo de QA de San Services decidió cubrir todas estas páginas de manera automatizada revisando ciertas secciones en específico de las páginas web como ser la disponibilidad y visualización de la imágenes, los precios, funcionalidad de los sliders y popup y el funcionamiento de los links.

4.2.1.2 Descripción General del Proyecto

El proyecto de QA Contents consiste en verificar el estado de los precios, visibilidad y disponibilidad de las imágenes, estado de los enlaces, funcionamiento de sliders y popup entre otros para la mayor parte de los sitios web de Sandals y Beaches.

El proyecto cubre un gran listado de páginas web que pertenecen a los sitios web de Sandals y Beaches en las siguientes ilustraciones de la 28 a la 34 se marcan cuáles fueron los sitios para los que se les crearon las pruebas automatizados.

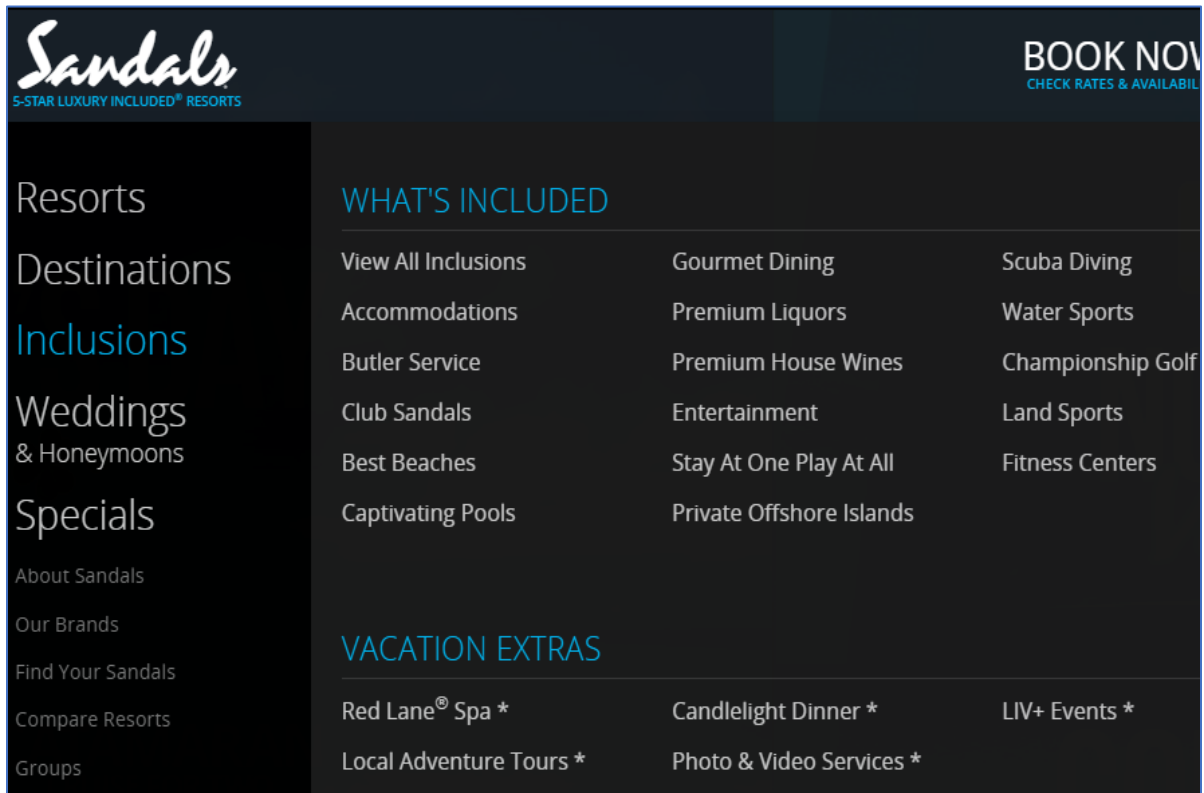


Ilustración 28: Páginas de Sandals Inclusions

Fuente: Propia

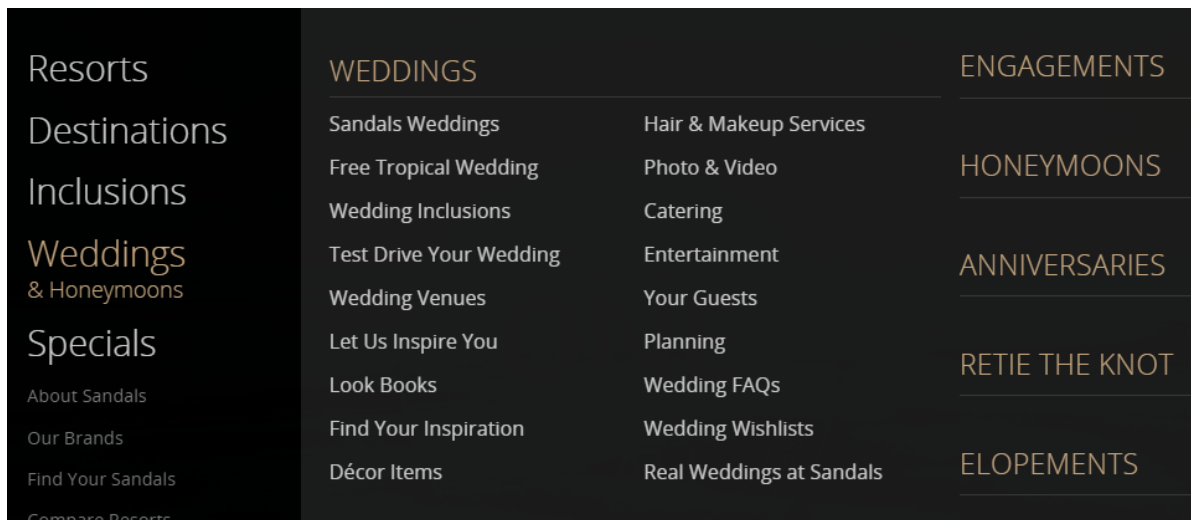


Ilustración 29: Páginas de Sandals Weddings

Fuente: Propia



Ilustración 30: Resto de sitios de Sandals

Fuente: Propia

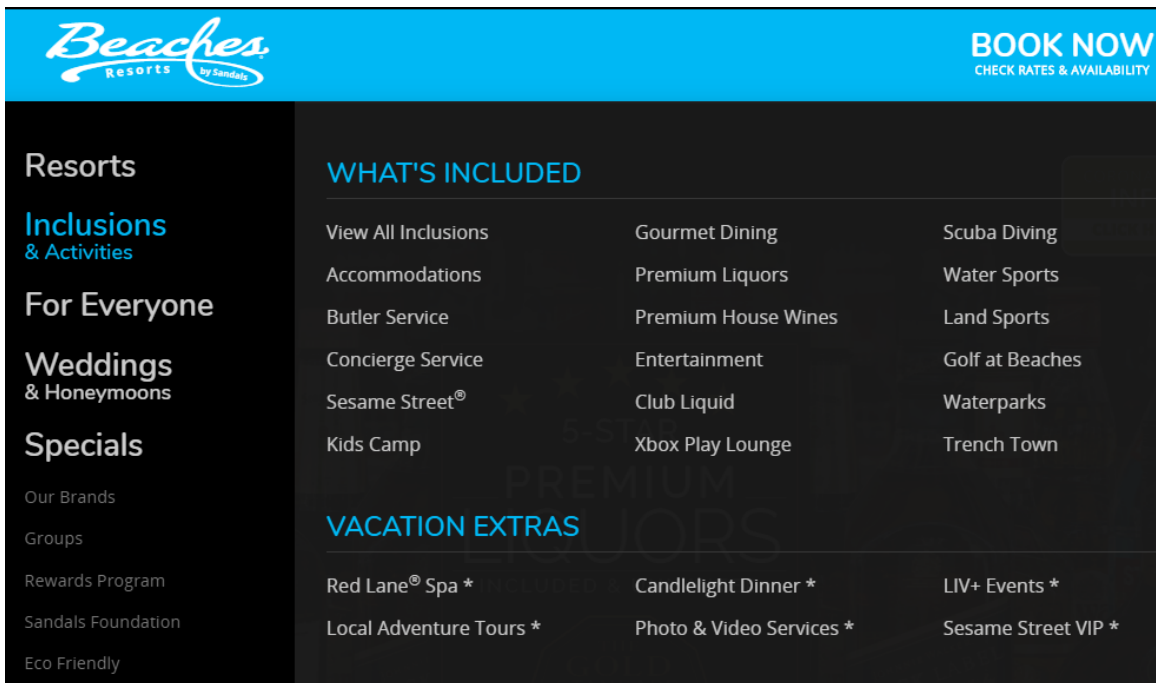


Ilustración 31: Páginas de Beaches Inclusions

Fuente: Propia

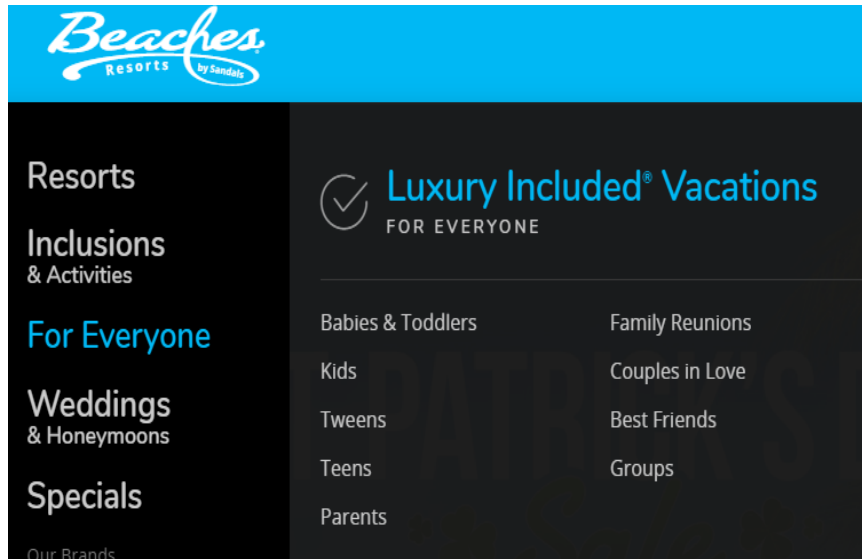


Ilustración 32: Páginas de Beaches For Everyone

Fuente: Propia

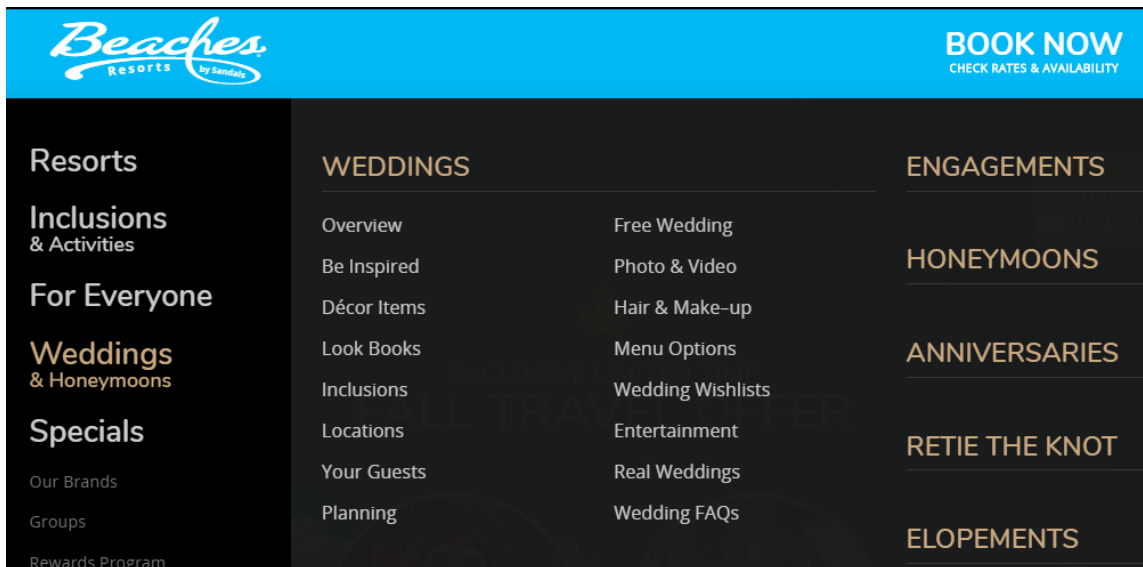


Ilustración 33: Páginas de Beaches Weddings

Fuente: Propia

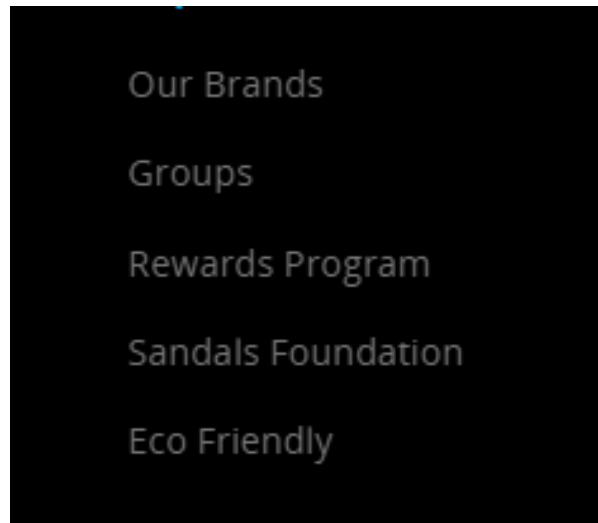


Ilustración 34: Resto de páginas de Beaches

Fuente: Propia

4.2.1.1 Bitácora de las Actividades Realizadas

La bitácora del trabajo realizado en este proyecto se presentará dividido en meses, en cada uno de los cuales se resumen las actividades destacadas que se realizaron a lo largo del desarrollo del proyecto y se concluirá con una bitácora diaria de las actividades realizadas por semana del mes.

Cabe añadir que después de la creación de los step objects necesarios para manipular ciertos elementos del DOM, el desarrollo del proyecto se torna repetitivo por el hecho de que la estructura de los specs y page objects vienen siendo similares con la única diferencia de que los selectores varían de página a página. Por lo que en algunos meses se omitirán ciertos aspectos similares para no hacer que el documento cuente con un exceso de imágenes.

4.2.1.2 Mes de octubre

Durante el tiempo restante del mes de octubre se trabajó principalmente en lo que es la estructura del proyecto y como se describe en la sección de capacitación todos los proyectos siguen la misma estructura y utilizan las mismas tecnologías por lo que al ser el inicio del proyecto solo se pudieron crear las dos principales brand object que son Sandals y Beaches y se creó el único country object enfocado en la región de Estados Unidos ya que el proyecto solo se enfocaría en

esa región. Además, se realizaron varias reuniones en las cuales se determinó que sitios web de Sandals y Beaches tendrían pruebas automatizadas y que se revisaría en cada una de ellas llegando a la conclusión que el proyecto revisaría únicamente el estado de las imágenes, links y precios presentes en todas las páginas web de Sandals y Beaches. En las ilustraciones 28 a la 34 se puede observar cuales fueron las páginas web que serían automatizadas.

En la siguiente semana se comenzó a identificar que selectores serían necesarios para la manipulación de las imágenes y los enlaces (Ilustraciones 35 y 36) de la página de Accommodation y en base a eso se creó el page object para la página.

```
/**Accomodation Selectors Images */
this.aiAccommodationImgsSelector = 'body .main img:not(.slick-slide)';
this.aiAccommodationSlidersSelectors = {
  lovenest:      'body .fou__slider-wrap img:not(.slick-cloned)',
  tranquility:  'body .js-slick-slider-tranquility img:not(.slick-cloned)',
  bathrooms:    'body .js-slick-slider-bathrooms img:not(.slick-cloned)',
  fortwo:       'body .js-slick-slider-fortwo img:not(.slick-cloned)'
}
this.aiAccommodationCurrentSlideSelectors = {
  lovenest:      'body .fou__slider-wrap img:not(.slick-cloned).slick-current',
  tranquility:  'body .js-slick-slider-tranquility img:not(.slick-cloned).slick-current',
  bathrooms:    'body .js-slick-slider-bathrooms img:not(.slick-cloned).slick-current',
  fortwo:       'body .js-slick-slider-fortwo img:not(.slick-cloned).slick-current'
}
this.aiAccommodationNextSlideButtonSelectors = {
  lovenest:      'body .fou__slider-btn--next',
  tranquility:  'body .js-tranquility-next',
  bathrooms:    'body .js-bathrooms-next',
  fortwo:       'body .js-fortwo-next'
}
```

Ilustración 35: Selectores de las imágenes de la página de Accommodation

Fuente: Propia


```

this.aiAccommodationLinksSelector = [
  'body .lux-sanctuaries a',
  'body .nestsuites-main-link',
  'body .link-spirits',
  'body .link-fortwo',
  'body .link',
  'body .standards-blocks-inner.standards-blocks-inner_right a',
  'body .whats-included-main-link'
];

```

Ilustración 36: Selectores de los enlaces de la página de Accommodation

Fuente: Propia

Como se puede observar en las ilustraciones se obtienen los selectores específicos de cada enlace e imagen, esto se realiza de esta forma debido a que las páginas web que utilizan algún framework tiende a tener copias de las imágenes y enlaces que no están visibles en la página por lo que si se optaba por usar selectores que obtuvieran todas las imágenes y todos los enlace, los resultados estarían llenos de copias que no se pueden interactuar en la página web y lo cual conlleva a errores con WebDriverIO.

Al avanzar la semana se agregó una nueva función en el archivo wdio.config para que cuando un popup (Ilustración 37) apareciera este lo cerrara inmediatamente ya que impedía a WebDriverIO interactuar con las imágenes de Sandals Accommodation.

```

/**
 * Runs before a WebdriverIO command gets executed.
 * @param {String} commandName hook command name
 * @param {Array} args arguments that command would receive
 */
beforeCommand: function (commandName) {
  try{
    if(browser.isVisible('body .subSave')){
      browser.selectorExecute('body .subSave i[class*=close]', function(closeButton) { closeButton[0].click(); }
      browser.waitUntil(function(){ return !browser.isVisible('body .subSave'); });
    }
  }catch(e){}
},

```

Ilustración 37: Función para cerrar popup

Fuente: Propia

Se creo un step object para verificar la visibilidad y el estado de los enlaces (Ilustración 38), con estado me refiero a si redireccionaba a una página con un estado de "200" el cual indica que la pagina está disponible, por lo tanto, se instaló una librería de NodeJS llamada axios que es una herramienta para hacer request a los URL por lo tanto se creó una nueva función para hacer uso de esto (Ilustración 39).

```
jets > links.step.js > ...
import Step from './step';
import {request} from './helpers/request';
import {axios_request} from './helpers/axios_request';

class Links extends Step {
  constructor(elementsSelector){
    super();
    this.elementsSelector = elementsSelector;
  }
  get links() {return browser.elements(this.elementsSelector)}



  linksAreLoaded(){
    let assertionValue = browser.waitUntil(() => this.links.value.every((link) => link.isVisible())&& this.links.value.length > 0);
    let urls = browser.execute((links) => links.value.map((link) => link.href), this.links);
    let urlsOnly = urls.value;
    let unique = (urlsOnly) => urlsOnly.filter((v, i) => urlsOnly.indexOf(v) === i);
    urls = unique(urlsOnly);
    let requests = axios_request(...urls);
    let badLinks = [];
    let badStatus = [];
    let assertionMessage = '';
    if(request === undefined){
      assertionMessage = 'Request is undefined. Please check if urls array is not empty.';
      return {assertionValue, assertionMessage};
    }
    requests.map((request, index) => {
      if(request !== 200){
        assertionValue = false;
        badLinks.push(index+" "+ urls.value[index]);
        badStatus.push(request);
      }
    });
    assertionMessage = `This links has problems. \nLinks: < ${badLinks} > \nStatus: < ${badStatus} >`;
    return {assertionValue, assertionMessage};
  }
  click(index){
    this.links.value[index].click();
  }
  scroll(index){
    if(index === undefined){
      let locationList = this.links.value.map((link) => link.getLocation().y);
      locationList.forEach((location) => {
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF

Ilustración 38: Step object para los enlaces

Fuente: Propia

```

rs >  axios_request.js >  axios_request
const axios = require('axios');

export function axios_request(...urls){
  const requests = urls.map(url => axios.get(url));
  let result = browser.call(() => {
    return axios.all(requests)
      .then(responses => {
        return responses.map((res) => res.status);
      }).catch((error) => {
        if(error.response.status === 500){
          return [error.response.status, error.response.data];
        }
        return [error.response.status];
      });
  });
  return result;
}

```

Ilustración 39: Función para realizar request

Fuente: Propia

Una vez realizado el step object para los enlaces se procedió a crear dos nuevos, uno para manipular los botones (Ilustración 40) y otro para las imágenes (Ilustración 41) los cuales se aseguran de que ambos elementos estén visibles en la página de Sandals Accommodation y en el caso de que las imágenes estén visibles realiza una verificación extra que es determinar si no es una broken image (imagen rota en español).

```

import Step from './step';
import {request} from './helpers/request';

class Images extends Step {
  constructor(elementsSelector){
    super();
    this.elementsSelector = elementsSelector;
  }
  get images() { return browser.elements(this.elementsSelector) }
  get length() { return this.images.value.length }
  imagesAreLoaded(){
    let assertionMessage, assertionValue, requests, src;
    let urls = [];
    assertionValue = browser.waitUntil(() => this.images.value.every((img) => {
      return img.isVisible();
    }) && this.images.value.length > 0);
    let locationList = this.images.value.map((image) => image.getLocation().y);
    locationList.forEach((location, index) => {
      browser.scroll(0, location)
      src = this.images.value[index].getAttribute('src');
      if(browser.desiredCapabilities.browserName == 'firefox' && !src.match('https:')){
        src = `https:${src}`;
      }
      urls.push(src);
    });
    if(urls.length === 0){
      assertionValue = false;
      assertionMessage = `The '${this.elementsSelector}' selector can't be loaded.`
      return {assertionValue, assertionMessage};
    }
    requests = request(...urls);
    requests.map((req) => {
      if(req['status'] !== 200){
        assertionValue = false;
        assertionMessage = `The image with the URL <{ ${req['url']} }> cant be loaded`;
        return {assertionValue, assertionMessage};
      }
    });
    return {assertionValue, assertionMessage};
  }
}

```

Ilustración 40: Step object para manipular las imágenes

Fuente: Propia

Ademas se agregaron nuevas características a los steps objects de links e imágenes para poder obtener el atributo src y agregarle el protocolo "https" porque cuando los specs son ejecutados en el navegador Firefox el src no tiene el protocolo, y por ende esto generaba errores.

```

import Step from './step';

class Button extends Step {

  constructor(elementSelector){
    super();
    this.elementSelector = elementSelector;
  }

  get button() { return browser.element(this.elementSelector); }
  get buttons() { return browser.elements(this.elementSelector); }
  get length() { return this.buttons.value.length; }
  click(){
    let assertionMessage = `This selector can't be clicked.`, assertionValue = false;
    if(browser.waitUntil(() => this.button.isVisible())){
      this.button.click();
      assertionValue = true;
    }
    return {assertionValue, assertionMessage};
  }
  scroll(x = 0, y = 0){
    if(browser.waitUntil(() => this.button.isVisible()))
      if(x!==0 || y !== 0){
        browser.scroll(this.elementSelector, x, y);
      }else{
        this.button.scroll();
      }
  }
  clickRandom(index = Math.floor(Math.random() * (this.buttons.value.length))){
    browser.waitUntil(() => this.buttons.value.every((button) => button.isVisible()) && this.buttons.value.length > 0);
    let locationList = this.buttons.value.map((button) => button.getLocation().y);
    browser.scroll(0, locationList[index]-200);
    this.buttons.value[index].click();
  }
  isExisting(){
    return this.button.isExisting();
  }
  back(){
    browser.back();
  }
  close(){
    browser.close();
  }
}

```

Ilustración 41: Step object para manipular los botones

Fuente: Propia

Tabla 4: Actividades realizadas Semana 3

Fecha	Descripción
Lunes 21 de octubre del 2019	Fui asignado al proyecto nuevo proyecto "QA Contents". Como el proyecto era nuevo se realizó una reunión para determinar qué acciones serían automatizadas.
Martes 22 de octubre del 2019	Participo en otra reunión en la que se determinó que sitios web de Sandals y Beaches tendrían test automatizados.
Miércoles 23 de octubre del 2019	Participo en otra reunión en la que se determinó que el proyecto revisaría únicamente el estado de las imágenes, links y precios presentes en todas las páginas de Sandals y Beaches que se habían determinado el día anterior.
Jueves 24 de octubre del 2019	Comencé a generar la estructura del proyecto y determinar cuáles serían los page object para cada una de las páginas web de Sandals y Beaches. Al final del día solo genere dos brand objects para las páginas principales.
Viernes 25 de octubre del 2019	Participo en una reunión en la que se explicó información general sobre los resorts de Sandals y Beaches.

Fuente: Propia

Tabla 5: Actividades realizadas Semana 4

Fecha	Descripción
Lunes 28 de octubre del 2019	Comencé a agregar los selectores de imágenes y links presentes en la página de Sandals Accommodation a su respectivo page object y a crear el primer spec del proyecto. Además, Se creo el repositorio de "QA Contents" para subir todo lo realizado hasta el momento.
Martes 29 de octubre del 2019	Se agrego nuevas funciones en el wdio.config para evitar la aparición de un popup que impedía a WebDriverIO interactuar con los botones de Sandals All Inclusive, se agregó la librería de axios para poder verificar el estado de los enlaces y se creó un step object para la manipulación de estos.
Miércoles 30 de octubre del 2019	Se creo un step object para manipular los botones y otro para las imágenes en el cual se verifica si las imágenes están visibles en la página de Sandals Accommodation y si en caso de estar visibles no sean broken image (imágenes rotas en español).
Jueves 31 de octubre del 2019	Se agregaron nuevas características a los steps objects de links e imágenes para poder obtener el atributo src y agregarle el "https" ya que en Firefox al obtener este atributo de las imágenes y links este los devolvía sin el protocolo por lo tanto al pedir un request al src este devolvía un error.

Fuente: Propia

4.2.1.3 Mes de noviembre

Durante este mes se trabajó principalmente en la creación de specs, en añadir nuevos selectores a la brand object de Sandals y en la creación de nuevos page object para los demás sitios web.

Comenzaremos con la creación y finalización del spec para la página de Sandals Accommodation (Ilustración 42) el cual una vez creados los step objects necesarios para manipular las imágenes, enlaces y botones su resolución fue rápida.

```
describe('Should check if the Accommodation page charge well.', function(){
  let us = new US();
  let sandals = new Sandals(us);
  let accommodation = new Accommodation(sandals);
  let assertion;
  let sliders = ['lovenest', 'tranquility', 'bathrooms', 'fortwo'];
  it('Should open ${accommodation.name} page', function(){
    accommodation.open();
    assertion = accommodation.waitLoading();
    expect(assertion.assertionValue, assertion.assertionMessage).to.be.true;
  });

  it('Should check if the image of Unique Suites show well.', function(){
    let imgs = new Images(accommodation.ImgsSelector);
    assertion = imgs.imagesAreLoaded();
    expect(assertion.assertionValue, assertion.assertionMessage).to.be.true;
  });

  it('Should check if the images in the Accommodation Sliders shows well.', function(){
    let accommodationSlidesButton, accommodationSlides, accommodationCurrentSlide;
    sliders.map((slider) => {
      accommodationSlidesButton = new Button(accommodation.NextSlideButtonSelectors[slider]);
      accommodationSlides = new Images(accommodation.SlidersSelectors[slider]);
      for(let i=0; i< accommodationSlides.length; i++){
        accommodationCurrentSlide = new Images(accommodation.CurrentSlideSelectors[slider]);
        assertion = accommodationCurrentSlide.imagesAreLoaded();
        expect(assertion.assertionValue, assertion.assertionMessage).to.be.true;
        accommodationSlidesButton.click();
      }
    });
  });
});
```

Ilustración 42: Spec de Sandals Accommodation

Fuente: Propia

Continuamos con la asignación de nuevos selectores a la brand object de Sandals (Ilustración 43), con la creación de un nuevo page object para la página de Butler Services (Ilustración 44) y a la creación de su respectivo spec (Ilustración 45).

```
/**Butler Selectors Images */
this.aiButlerPath = 'butler';
this.aiButlerImgsSelector = 'body .main img:not([alt="butler logo"])';
this.aiButlerDropdownSelector = 'body .dropdown-resorts';
this.aiButlerDropdownItems = 'body .ui-menu-item';

/**Butler Selector Link */
this.aiButlerMomentsLinkSelector = 'body .english-butler__btn'
```

Ilustración 43: Selectores para la página de Butler Services

Fuente: Propia

```

import Page from './page';

class Butler extends Page {
  constructor(brand){
    super(brand);
    this.name = 'Butler';
    this.Path = this.brand.aiButlerPath;
    /**Sandals/Beaches Selectors Images */
    this.ImgsSelector = this.brand.aiButlerImgsSelector;
    this DropdownSelector = this.brand.aiButlerDropdownSelector;
    this DropdownItems = this.brand.aiButlerDropdownItems;
    /**Sandals Selector Link/Image */
    this.MomentsLinkSelector = this.brand.aiButlerMomentsLinkSelector;
    /**Beaches Selectors */
    this.SuitesSelector = this.brand.aiButlerSuitesSelector;
    this.SelectorPage = 'body .butler';
  }

  get page() {
    return browser.element(this.SelectorPage);
  }

  open(){
    super.open('all-inclusive', this.Path); //butler-service';
  }
}

```

Ilustración 44: Page object de la página Butler Services

Fuente: Propia

```

describe('Should check if the Butler page charge well.', function(){
  let us = new US();
  let sandals = new Sandals(us);
  let butler = new Butler(sandals);
  let assertion;

  it(`Should open ${butler.name} page`, function(){
    butler.open();
    assertion = butler.waitLoading();
    expect(assertion.assertionValue, assertion.assertionMessage).to.be.true;
  });

  it('Should check if the images of Butler page show well.', function(){
    let butlerEliteImg = new Images(butler.ImgsSelector);
    assertion = butlerEliteImg.imagesAreLoaded();
    expect(assertion.assertionValue, assertion.assertionMessage).to.be.true;
  });

  it('Should check if the English Butler link work well.',function(){
    let nestLinks = new Links (butler.MomentsLinkSelector);
    assertion = nestLinks.linksAreLoaded();
    expect(assertion.assertionValue, assertion.assertionMessage).to.be.true;
  });
});

```

Ilustración 45: Spec de la página Butler Services

Fuente: Propia

Durante el desarrollo de este spec y la creación de dos nuevos page objects para las páginas de Best Beaches y Sandals Club (Ilustración 46), se realizó una reunión para determinar que otros aspectos debería de verificar las pruebas automatizadas del proyecto de Contents, por lo que durante dicha reunión se observó que los sliders presentes en algunas páginas deberían de ser verificados por lo tanto se procedió a crear un spec object que los manipulara (Ilustración 47).

```
import Page from './page';

class BestBeaches extends Page {
  constructor(brand){
    super(brand);
    this.name = 'Best Beaches';

    /**Sandals Selectors Images */
    this.ImgsSelector = this.brand.aiBestBeachesImgsSelector;
    this.SliderElementsSelectors = this.brand.aiBestBeachesSliderElementsSelectors;
    this.PopupButtonsSliderSelectors = this.brand.aiBestBeachesPopupButtonsSliderSelectors;
    /**Sandals Selectors Links */
    this.ResortLinkSelector = this.brand.aiBestBeachesResortLinkSelector;
    this.ViewLinkSelector = this.brand.aiBestBeachesViewLinkSelector;
  }

  get page() {
    return browser.element('body .best-beaches');
  }

  open(){
    super.open('all-inclusive/best-beaches/');
  }
}

import Page from './page';

class ClubSandals extends Page {
  constructor(brand){
    super(brand);
    this.name = 'Club Sandals';

    /**Sandals Selectors Images */
    this.ImgsSelector = this.brand.aiClubSandalsImgsSelector;
    this.DropDownSelectors = this.brand.aiClubSandalsDropDownSelectors;
  }

  get page() {
    return browser.element('body .sandals-club');
  }

  open(){
    super.open('all-inclusive/sandals-club/');
  }

  waitLoading(){
    return super.waitLoading(this.page);
  }
}
```

Ilustración 46:Page objects de Best Beaches y Sandals Club

Fuente: Propia

La creación de dicho step object tomo mucho tiempo aproximadamente 4 horas debido a que los sliders tienden a generar copias de sus imágenes ya que estos son como ciclos de imágenes que sin importar cuantas veces des clic en el botón de siguiente este nunca dejara de mostrar imágenes por lo que encontrar selectores que fijaran la imagen actual resulto una tarea algo complicado debido a mi poca experiencia en el manejo de selectores.

```

class Slider {
  constructor(sliderElements, amountOfImagesPerSlide) {
    this.sliderElements = sliderElements;
    this.amountOfSlides = getAmountOfElements(this.sliderElements.imagesSelector) / amountOfImagesPerSlide;
  }
  /**
   * This function check all the images, popups and links of a slider
   * @param {Boolean} haveLinks - only if have links
   * @param {Boolean} navByIndex - only if have navigation buttons instead of arrow buttons
   * @param {Boolean} havePopup - only if have popups
   * @param {Boolean} popupIconImage - only if have image in the popup
   * @param {Boolean} popupLinks - only if have links in the popup
   */
  checkSliderContent(haveLinks, navByIndex, havePopup, popupIconImage, popupLinks) {
    for (let index = 0;
         index < this.amountOfSlides;
         index++)
    {
      checkImagesHttpRequest(this.sliderElements.currentImageSelector, true);
      if(haveLinks)
        checkLinksHttpRequest(this.sliderElements.currentLinksSelector);
      if(havePopup) {
        clickElement(this.sliderElements.openPopupButtonSelector);
        checkIfElementLoads(this.sliderElements.popupSelector);
        if(popupIconImage)
          checkImagesHttpRequest(this.sliderElements.currentIconImageSelector, true);
        if(popupLinks)
          checkLinksHttpRequest(this.sliderElements.currentLinksSelector);
        clickElement(this.sliderElements.closePopupButtonSelector);
      }
      if(navByIndex){
        if(index+1 <= this.amountOfSlides-1){
          scrollPage(this.sliderElements.currentImageSelector);
          clickElement(this.sliderElements.nextButtonSelector, index+1);
        }
        browser.pause(1000);
      }
      else{
        scrollPage(this.sliderElements.currentImageSelector);
        clickElement(this.sliderElements.nextButtonSelector);
      }
    }
  }
}

```

Ilustración 47: Step object para manipular todo tipo de step object

Fuente: Propia

Una vez finalizado el step object se retornó a los specs de Accommodation para agregar esta nueva clase ya que esta página web cuenta con sliders que no estaban siendo verificados por las pruebas automatizadas.

Al día siguiente se realizó otro reunión para determinar si los popup también deberían de ser revisados en el proyecto de Contents. Al finalizar la reunión se concluyó que los popups también deberían de ser verificados en las pruebas por lo tanto se prosiguió a la creación de un nuevo step object para la manipulación de los popups (Ilustración 48).

```

import Step from './step';

class Popup extends Step {

  constructor(elementSelector){
    super();
    this.elementSelector = elementSelector;
  }

  get popup() {return browser.element(this.elementSelector)}

  popupIsLoaded(){
    let assertionValue, assertionMessage;
    try {
      assertionValue = browser.waitUntil(() => this.popup.isVisible());
      assertionMessage = ' ';
    } catch (error) {
      assertionValue = false;
      assertionMessage = `The popup wasnt displayed.`;
      return { assertionValue, assertionMessage};
    }
    return { assertionValue, assertionMessage};
  }
}

export default Popup;

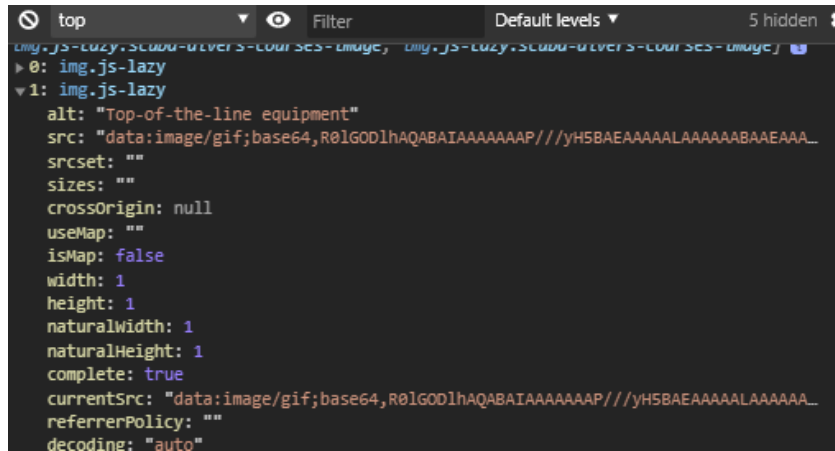
```

Ilustración 48: Step object para la manipulación de los popups

Fuente: Propia

Al final de la semana 5 se terminaron realizando varios specs y page object para las páginas de Sandals Gourmet Dining, Captivating Pools, Offshore Islands, Multi Destinations, Entertainment y Mondavi Wines.

Continuando con la semana 6 se trabajó en la creación de un nuevo page object y spec para la página de Scuba Diving en la creación del spec se presentaron varios problemas con el step object que manipulaba las imágenes esto se debía a que las imágenes en la página de Scuba Diving tiene un efecto CSS denominado lazy image (imagen lenta) el cual hace que las imágenes carguen en su totalidad hasta que se está sobre ellas por lo cual presentaba un error al momento de revisar el src de estas imágenes (Ilustración 49) por que al no estar cargadas estas daban un src erróneo al cual no se le podía hacer request.



```
img.js-lazy,scuba-ulvers-cour ses-umage, img.js-lazy,scuba-ulvers-cour ses-umage
> 0: img.js-lazy
▼ 1: img.js-lazy
  alt: "Top-of-the-line equipment"
  src: "data:image/gif;base64,R0lGODlhAQABAIAAAAAAAP///yH5BAEAAAAALAAAAAABAAEAAA...
  srcset: ""
  sizes: ""
  crossOrigin: null
  useMap: ""
  isMap: false
  width: 1
  height: 1
  naturalWidth: 1
  naturalHeight: 1
  complete: true
  currentSrc: "data:image/gif;base64,R0lGODlhAQABAIAAAAAAAP///yH5BAEAAAAALAAAAA...
  referrerPolicy: ""
  decoding: "auto"
```

Ilustración 49: Src de una lazy-image que no a termina de cargar

Fuente: Propia

En el resto de la semana se crearon nuevos page objects y specs para las páginas de New Drive, Certified, Inclusions, Fleet, Dive Sites, Scuba Courses y Tours. En este último spec se realizó una reunión el viernes para determinar si se revisarían los sliders y enlaces generados por el dropdown de resorts. Al final de la reunión se terminó descartando esta verificación ya la cantidad de sliders y enlaces generados eran demasiados y ampliaban en gran medida la duración del spec lo cual no aportaba ningún beneficio por lo tanto se descartó esta verificación.

Continuando con el trabajo realizado durante la semana 7 en esta se crearon nuevos page objects y specs para las páginas de Candlelight Dinner, Photo And Video, Sandals Liv Plus y se agregaron nuevos selectores a los page objects y nuevas pruebas dentro de los specs de Captivating Pools y Sandals Club, ya que se creó un nuevo step object que manipula los dropdowns (Ilustración 50). Se realizo una reunión en la que se aprobó que el contenido desplegado al darle clic a un checkbox debería de ser revisado. Por lo que se creó un nuevo step object que se encargara de verificar la visibilidad y el funcionamiento de los checkboxes (Ilustración 51).

```

class Dropdown extends Step {
  constructor(elementSelector, elementsSelector, contentSelector = undefined){
    super();
    this.elementSelector = elementSelector;
    this.elementsSelector = elementsSelector;
    this.contentSelector = contentSelector;
    this.selectedOption = '';
  }
  get dropdown() {
    return browser.element(this.elementSelector);
  }
  get dropdowns(){ return browser.elements(this.elementSelector); }
  get dropdownElements() {return browser.elements(this.elementsSelector);}
  get length() {return this.dropdownElements.value.length; }
  get option() {return this.selectedOption; }
  get content() {return this.contentSelector; }
  chooseOption(option = Math.floor(Math.random() * (this.length - 1) + 1) ){
    let assertionMessage = `The <${this.dropdown}> element never shows up.`;
    let assertionValue = false;
    if(browser.waitUntil(() => this.dropdown.isVisible())) {
      if(browser.waitUntil(() => this.dropdownElements.isVisible())){
        let element = this.dropdownElements.value[option];
        this.selectedOption = element.getText();
        if(browser.desiredCapabilities.browserName == 'firefox' && this.content !== undefined){
          browser.execute((e) => e.click(), element);
          browser.waitUntil(() => browser.isVisible(this.content));
          browser.waitUntil(() => browser.isVisible(this.content));
          browser.execute((e) => e.click(), element);
        }else{
          element.click();
        }
      }
      let requests = request(browser.getUrl());
      let badLinks = [];
      assertionValue = true;
      requests.forEach(function(request){
        if(request['status']!=200){
          assertionValue = false;
          badLinks.push(request['url']);
        }
      });
      assertionMessage = `This option <${this.selectedOption}> of the dropdown returns a 404 page not found error.`;
      return {assertionValue, assertionMessage}
    }
  }
}

```

Ilustración 50: Step object que manipula los dropdowns

Fuente: Propia

```

1  import Step from './step';
2
3  class CheckBox extends Step {
4
5      constructor(elementSelector){
6          super();
7          this.elementSelector = elementSelector;
8      }
9      get check() {return browser.elements(this.elementSelector);}
10     get length(){
11         return this.check.value.length;
12     }
13     clickToCheckRandomly(reduce_range = 0){
14         let option = Math.floor(Math.random() * (this.length - reduce_range))
15         let visible = browser.waitUntil(() => this.check.value.map((box) => {
16             box.isVisible()
17             box.scroll();
18         }));
19         let val = browser.execute((cbxs, opt) => cbxs.value[opt].click() ,this.check, option);
20     }
21     scroll(x = 0, y = 0){
22         browser.scroll(this.elementSelector[0], x, y);
23     }
24     back(){
25         browser.back();
26     }
27     close(){
28         browser.close();
29     }
30     switchTab(){
31         browser.switchTab();
32     }
33 }
34 export default CheckBox;

```

Ilustración 51: Step object que manipula los checkboxes

Fuente: Propia

Al final de la semana se realizaron varias modificaciones al step object encargado de manipular las imágenes por el hecho que en ciertas paginas estas tardaban en aparecer por ser lazy-image, anteriormente se había creado una solución pero en ciertas paginas esta solución no era tan efectiva por lo que se optó por inspeccionar toda la página antes de obtener el src de las imagines, esta nueva solución resulto ser muy efectiva porque no se volvieron a presentar problemas con este tipo de imágenes. Al final se realizó un cambio en los selectores de la página de Gourmet Dining ya que se agregaron nuevas imágenes.

Continuando con el trabajo realizado en la semana 8 se finalizó con todos los specs y page objects pertenecientes a Sandals Inclusions. Con esta sección ya realizada se procedió a crear los page objects y specs para las páginas de Beaches Inclusions, también se llevó a cabo una reunión para cambiar la librería de node-fetch por axios en todo el proyecto ya que muestra más información

detallada a comparación de la antes mencionada. Además, cabe añadir que se cambiaron varios describes y its para hacerlos más fáciles de digerir para los usuarios que leen los reportes.

Al finalizar la semana se crearon nuevos page objects y specs para la sección de Beaches Inclusions la cuales son Accommodation, Concierge Services, Sesame Street, Kids Camp, Dining, Premium Liquors, Robert Mondavi, Entertainment y Club Liquid.

Tabla 6: Actividades realizadas Semana 5

Fecha	Descripción
Viernes 1 de noviembre del 2019	Se terminaron las pruebas para la página de Sandals Accommodation y se creó un nuevo page object con todos los selectores y atributos y un nuevo spec para la página de Sandals Butler Services.
Lunes 4 de noviembre del 2019	Se termino el spec para la página de Sandals Butler Services y se crearon dos nuevos page objects y specs para las páginas de Sandals Best Beaches y Sandals Club.
Martes 5 de noviembre del 2019	Se realizo una reunión para determinar si sliders también serian agregados a el proyecto de QA Contents lo cual se aprobó. Se termino el spec de Sandals Club y se creó un nuevo step object para manipular sliders presentes en la página de Sandals Best Beaches y Accommodation.
Miércoles 6 de noviembre del 2019	Se termino el spec para la página de Sandals Best Beaches y se crearon nuevos page objects y specs para las páginas de Sandals Gourmet Dining y Captivating Pools.
Jueves 7 de noviembre del 2019	Se realizo una reunión para determinar si los popup también deberían de ser revisados en el proyecto de QA Contents el cual se aprobó. Se creo un nuevo step object para la manipulación de los popup y se terminaron los para Sandals Gourmet Dining y Captivating Pools.
Viernes 8 de noviembre del 2019	Se crearon nuevos page objects para las páginas de Sandals Offshore Islands, Multi Destinations, Entertainment y Mondavi Wines. Para los cuales también se crearon sus respectivos specs.

Fuente: Propia

Tabla 7: Actividades realizadas Semana 6

Fecha	Descripción
Lunes 11 de noviembre del 2019	Se creo un nuevo page object para la página de Sandals Scuba Diving y se creó su respectivo spec. También se añadieron nuevas funciones para el step object de imágenes ya que las imágenes en la página de Scuba Diving tiene un efecto CSS denominado lazy image (imagen lenta) el cual denominado efecto hace que las imágenes carguen en su totalidad hasta que se está sobre ellas por lo cual presentaba un error al momento de revisar el src de estas imágenes ya que al no estar cargadas estas daban un src erróneo al cual no se le podía hacer request.
Martes 12 de noviembre del 2019	Se crearon nuevos page objects y specs para las subpáginas de Sandals Scuba Diving las cuales son New Drive, Certified, Inclusions, Fleet, Dive Sites y Scuba Courses.
Miércoles 13 de noviembre del 2019	Se crearon nuevos page objects y specs para las páginas de Sandals Championship Golf, Landsports, Fitness Center y Redlane Spa.
Jueves 14 de noviembre del 2019	Se creo un nuevo page object y spec para la página de Sandals Tours.
Viernes 15 de noviembre del 2019	Se realizo una reunión para discutir las secciones que deberían de revisarse en la página de Sandals Tours ya que esta presento varios problemas al momento de crear test de las imagines y links porque al seleccionar un resort en el dropdown este desplegaba una gran cantidad de sliders de imágenes y links por lo que la prueba tardaba aproximadamente 8 minutos sección seria descartada.

Fuente: Propia

Tabla 8: Actividades realizadas Semana 7

Fecha	Descripción
Lunes 18 de noviembre del 2019	Se crearon nuevos page objects y specs para las páginas de Sandals Candlelight Dinner, Photo And Video.
Martes 19 de noviembre del 2019	Se creo un nuevo page object y un nuevo spec para la página de Sandals Liv Plus y se agregaron nuevos selectores a los page objects y nuevas pruebas dentro de los specs de Sandals Captivating Pools y Sandals Club.
Miércoles 20 de noviembre del 2019	Se realizo una reunión en la que se aprobó que el contenido desplegado al darle clic a un checkbox debería de ser revisado. Por lo que se creó un nuevo step object que se encargara de manipular la acción de darle clic a un checkbox.
Jueves 21 de noviembre del 2019	Se modifíco el step object encargado de manipular las imágenes ya que en algunas páginas de Sandals las imágenes tardaban en aparecer, por lo tanto, se añadió una función que hace que el navegador se moviera de manera dinámica a la posición en la que estarán las imágenes para que estas carguen primero antes de obtener el src.
Viernes 22 de noviembre del 2019	Se modifíco el step object encargado de manipular los popup, se le añadió más tiempo de espera ya que estos en ciertas páginas de Sandals tardan en aparecer una vez presionado el botón que los acciona.

Fuente: Propia

Tabla 9: Actividades realizadas Semana 8

Fecha	Descripción
Lunes 25 de noviembre del 2019	Se realizo una reunión para aprobar el cambio de node-fetch por la librería de axios ya que esta devuelve mensajes de error más comprensibles. Por lo tanto, las secciones de request de node-fetch para los steps objects que manipulan las imágenes y los enlaces fueron cambiados por los request de axios.
Martes 26 de noviembre del 2019	Se terminaron todos los page objects y specs para las páginas de Sandals Inclusions, además se cambiaron muchas definiciones de los describes y its de los specs. Con esta sección ya realizada se procedió a crear los page objects y specs para las páginas de Beaches Inclusions.
Miércoles 27 de noviembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Accommodation y Concierge Services.
Jueves 28 de noviembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Sesame Street, Kids Camp, Dining y Premium Liquors.
Viernes 29 de noviembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Robert Mondavi, Entertainment y Club Liquid.

Fuente: Propia

4.2.1.4 Mes de diciembre

Durante este mes se continuo con la creación de specs y page objects para las páginas de Beaches Scuba Diving, Water Sports, Land Sports y Beaches Golf. Waterparks, Trench Town y Redlane Spa. Al final de la semana 9 se terminaron todos los specs para las páginas de Beaches Inclusions y se inició con la creación de los specs y page objects las páginas web de Weddings Overview (Ilustración 52-53) y Weddings Inspiration de la sección de Sandals Weddings el resto de page objects y specs siguen la misma estructura de Weddings Overview.

```

class WeddingsOverview extends Page {
  constructor(brand){
    super(brand);
    this.name = 'Weddings Overview';
    this.Path = this.brand.weddingsOverviewPath;
    this.SelectorPage = this.brand.weddingsOverviewSelectorPage;
    this.ImgsSelector = this.brand.weddingsOverviewImgsSelectors;
    this.SliderSelector = this.brand.weddingsOverviewSliderSelector;
    this.CurrentSliderSelector = this.brand.weddingsOverviewCurrentSlideSelector;
    this.NextSlideButtonSelector = this.brand.weddingsOverviewNextSlideButtonSelector;
    this.LinksSelector = this.brand.weddingsOverviewLinksSelector;
  }

  get page() {
    return browser.element(this.SelectorPage);
  }

  open(){
    super.open(this.Path);
  }
}

```

Ilustración 52: Page object de Weddings Overview

Fuente: Propia

```

> sandals-us-imgs-links > weddings_overview_content.spec.js > ...
describe('Should check if the WeddingsOverview page load well.', function(){
  let us = new US();
  let sandals = new Sandals(us);
  let overview = new WeddingsOverview(sandals);
  let assertion;
  let runInclusiveTest = false;
  it('Should open ${overview.name} page', function(){
    overview.open();
    assertion = overview.waitLoading();
    runInclusiveTest = assertion.assertionValue;
    expect(assertion.assertionValue, assertion.assertionMessage).to.be.true;
  });

  it('Should check if the image of Weddings Overview are visible.', function(){
    if(!runInclusiveTest)
      this.skip();
    let imgs;
    overview.ImgsSelector.forEach((selector) => {
      imgs = new Images(selector);
      assertion = imgs.imagesAreLoaded();
      if(!assertion.assertionValue)
        runInclusiveTest = assertion.assertionValue;
      expect(assertion.assertionValue, assertion.assertionMessage).to.be.true;
    });
  });
});

```

Ilustración 53: Spec de Weddings Overview

Fuente: Propia

Durante la semana 10 se continuaron añadiendo selectores a él brand de Sandals y se crearon nuevos specs y page objects para las páginas de Decor, Weddings Inclusions, Weddings Look Books Ceremony, Reception, Your Guests, Planning, Free Weddings, Photo And Video, Menu Options, Entertainment, Honeymoon, Anniversary y Retie The Knot. Al finalizar la semana se completaron todos los specs de Sandals Weddings y se comenzó con el desarrollo de los specs y page objects para las páginas web de la sección de Beaches Weddings.

Durante la semana 11 se continuaron se empezó a añadir más selectores a la brand de Beaches y se crearon nuevos specs y page objects para las páginas de Weddings Overview, Be Inspired, Decor Ítems, Weddings Look Book, Locations, Inclusions, Your Guests, Planner, Free Weddings, Photo and Video, Weddings Menu, Entertainment, Honeymoon, Anniversary, Vow Renewals y Elopements.

Al final de la semana se terminaron todos los page objects y specs para las páginas de Beaches Weddings. Con esta sección finalizada se procedió a crear los page objects y specs para las páginas de Beaches For Everyone.

Durante la semana 12 se agregaron nuevos selectores a la brand de Beaches y se crearon nuevos page objects y specs para las páginas de Beaches For Everyone Toddlers, Kid, Tweens, Teens, Parents, Couples, Reunions, Best Friends y Groups. El viernes se realizaron cambios en las páginas de Beaches y Sandals Weddings por las promociones de año nuevo por lo que se trabajó en el cambio de selectores en los brand objects de Sandals y Beaches. Además, queda añadir que se terminaron todos los page objects y specs para las páginas web de la sección de Beaches For Everyone.

Durante la semana 13 se realizó una pequeña reunión determinar que nuevas paginas tendrán pruebas automatizadas. Además, se agregó un nuevo step object para los precios (Ilustración 54) y verificar que no sean menor o igual a 0 dólares y que estén visibles en ciertas páginas de Sandals y Beaches Inclusions y en algunas páginas de Weddings.

```

class Prices extends Step {
  constructor(elementSelector, couple = false){
    super();
    this.elementSelector = elementSelector;
    this.couple = couple;
  }
  get prices() {return browser.elements(this.elementSelector[0]);}
  get names() {
    if(this.couple){
      const original = browser.elements(this.elementSelector[1]);
      const copy = browser.elements(this.elementSelector[1]);
      const duplicateNames = original.value.concat(copy.value);
      return duplicateNames.sort((a,b) => {
        if ( a.index < b.index ){
          return -1;
        }
        if ( a.index > b.index ){
          return 1;
        }
        return 0;
      });
    }
    return browser.elements(this.elementSelector[1]).value;
  }
  priceIsLoaded(){
    let assertionValue = true;
    let assertionMessage = '';
    assertionValue = browser.waitUntil(() => this.prices.value.every((price ,index) =>{
      return price.isVisible();
    }) && this.prices.value.length > 0);
    if(assertionValue){
      this.prices.value.map((price, index) =>{
        if(price.getText() === "$0"){
          assertionValue = false;
          assertionMessage = `The ${this.names[index].getText()} tour is displaying a price of $0`;
          return {assertionValue, assertionMessage};
        }
      });
      return {assertionValue, assertionMessage};
    }
  }
  assertionMessage = `The tour prices doesn't appear.`;
  return {assertionValue, assertionMessage};
}

```

Ln 2

Ilustración 54: Step object para manipular los precios.

Fuente: Propia

Tabla 10: Actividades realizadas Semana 9

Fecha	Descripción
Lunes 2 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Scuba Diving, Water Sports, Land Sports y Beaches Golf.
Martes 3 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Waterparks, Trench Town y Redlane Spa.
Miércoles 4 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Tours, Photo And Video y Liv Plus.
Jueves 5 de diciembre del 2019	Se terminaron todos los page objects y specs para las páginas de Beaches Inclusions.
Viernes 6 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Sandals Weddings Overview y Weddings Inspiration.

Fuente: Propia

Tabla 11: Actividades realizadas Semana 10

Fecha	Descripción
Lunes 9 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Sandals Weddings Decor, Weddings Inclusions y Weddings Look Books Ceremony y Reception.
Martes 10 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Sandals Weddings Your Guests, Planning y Free Weddings.
Miércoles 11 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Sandals Weddings Photo And Video, Menu Options y Entertainment.
Jueves 12 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Sandals Weddings Honeymoon, Anniversary y Retie The Knot.
Viernes 13 de diciembre del 2019	Se terminaron todos los page objects y specs para las páginas de Sandals Weddings. Con esta sección ya realizada se procedió a crear los page objects y specs para las páginas de Beaches Weddings.

Fuente: Propia

Tabla 12: Actividades realizadas Semana 11

Fecha	Descripción
Lunes 16 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Weddings Overview, Be Inspired, Decor Items y Weddings Look Book.
Martes 17 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Weddings Locations, Inclusions, Your Guests, Planner y Free Weddings.
Miércoles 18 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Weddings Photo And Video, Weddings Menu, Entertainment y Honeymoon.
Jueves 19 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches Weddings Anniversary, Vow Renewals y Elopements.
Viernes 20 de diciembre del 2019	Se terminaron todos los page objects y specs para las páginas de Beaches Weddings. Con esta sección ya realizada se procedió a crear los page objects y specs para las páginas de Beaches For Everyone.

Fuente: Propia

Tabla 13: Actividades realizadas Semana 12

Fecha	Descripción
Lunes 23 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches For Everyone Toddlers, Kid, Tweens y Teens.
Martes 24 de diciembre del 2019	Vacaciones por celebración de la Navidad.
Miércoles 25 de diciembre del 2019	Vacaciones por celebración de la Navidad.
Jueves 26 de diciembre del 2019	Se crearon nuevos page objects y specs para las páginas de Beaches For Everyone Parents, Couples, Reunions, Best Friends y Groups.
Viernes 27 de diciembre del 2019	Por cambios en las páginas de Beaches y Sandals Weddings se trabajó en el cambio de selectores en los Brand Objects y en los respectivos page objects de cada una de las páginas web de Weddings. Se terminaron todos los page objects y specs para las páginas de Beaches For Everyone.

Fuente: Propia

Tabla 14: Actividades realizadas Semana 13

Fecha	Descripción
Lunes 30 de diciembre del 2019	Se realizó una reunión para agregar más páginas web al proyecto de QA Contents. Además, se agregó un nuevo step object para los precios y verificar el estado de estos en ciertas páginas de Sandals y Beaches Inclusions y en algunas páginas de Weddings.
Martes 31 de diciembre del 2019	Vacaciones por celebración del Año Nuevo.

Fuente: Propia

4.2.1.5 Mes de enero

Durante este mes se continuó añadiendo nuevos selectores al brand de Beaches y con la creación de nuevos specs y page objects para las páginas de Rewards, Beaches Sandals Foundation, Our Brands y Eco Friendly. Se agregaron nuevos selectores al brand de Sandals y con la creación de nuevos specs y page objects para las páginas de Sandals About, Groups, Rewards Program, Sandals Foundation, Eco Friendly, What's New, Compare Resorts y Vacation Planner.

Se actualizaron varios selectores para las páginas de Gourmet Dining, Entertainment, Weddings Planner, Honeymoon, Retie The Knot, Scuba Diving y Water Sports en la brand de Sandals. Además, se agregó la función de tomar screenshots (Ilustración 55-56) cuando uno de los specs falla. Se añadió un nuevo step object para detectar cuando una página falla por un estado de 404 (Ilustración 57) ya que durante esta semana muchas de las pruebas fallaban por estos errores.

```

afterTest: async function (test) {
  if(!test.passed){
    const sourceDir = process.env['MACAW_SRC'];
    const {browserName} = browser.desiredCapabilities;
    const {name}= browser.desiredCapabilities;
    const today = new Date();
    const selector = browser.getScreenshotSelector().toString();
    browser.saveElementScreenshot(sourceDir + '/screenshots/' + browserName + name + '-' + test.title.split(' ').join('_') + today.getMilliseconds() +
  }
}

```

Ilustración 55: Función para tomar screenshots 1

Fuente: Propia

```

function defineScreenshotTarget(target){
  let selector = target;
  if(target.hasOwnProperty('selector'))
    selector = target.selector;
  browser.addCommand('getScreenshotSelector', () => selector, true);
}
function takeScreenshot(place){
  const sourceDir = process.env['SAN_WDIO_SRC'];
  const {browserName} = browser.desiredCapabilities;
  const {name}= browser.desiredCapabilities;
  const today = new Date();
  var filename = sourceDir + '/screenshots/' + browserName + name + '-' + today.getMilliseconds() + '-' + today.toDateString()+'.png';
  console.log('Screenshot taken: ', browserName + name + '-' + today.getMilliseconds() + '-' + today.toDateString()+'.png');
  browser.saveScreenshot(filename);
}

export {defineScreenshotTarget, takeScreenshot};

```

Ilustración 56: Función para tomar screenshots 2

Fuente: Propia

```

import Step from './step';
import {request} from '../helpers/request';
import {axios_request} from '../helpers/axios_request';
class PageURL extends Step {

  constructor(){
    super();
  }
  get pageURL() {return browser.getUrl()}

  pageURLisLoaded(){
    let requests = request(this.pageURL);
    let assertionValue = true;
    let badLinks = [];
    requests.forEach(function(request){
      if(request['status']!==200){
        assertionValue = false;
        badLinks.push(request['url']);
      }
    });
    let assertionMessage = `This URL <${this.pageURL}> returns a 404 page.`;
    return {assertionValue, assertionMessage};
  }

  pageStatusIsOk(){
    let requests = axios_request(this.pageURL);
    let assertionValue = true;
    requests.forEach(function(req){
      if(req!==200){
        assertionValue = false;
      }
    });
    let assertionMessage = `This page <${this.pageURL}> return a <${requests}> status.`;
    return {assertionValue, assertionMessage};
  }

  pageErrorExist(selectorError){
    return browser.isExisting(selectorError)
  }
}
export default PageURL;

```

Ilustración 57: Step object para detectar el estado 404

Fuente: Propia

Durante la semana 15 se agregaron nuevos selectores a la brand de Sandals y se crearon nuevos page objects y specs para las páginas de Lest Us Inspire You, Location y Test Drive Your Wedding. Se realizo una reunión para determinar que se realizarán specs y page objects para las páginas

que se encuentran en Sandals Resorts, Sandals Our Islands y Beaches Resorts. Se crearon nuevos specs y page objects para las páginas de Jamaica, Saint Lucia, Bahamas, Barbados, Antigua y Granada. Al final de la semana se realizó una reunión en la que se determinó que no se crearan specs ni page object para las páginas que están en la sección de Sandals Resorts por el momento y se crearon nuevos page objects y specs para las páginas de Beaches Resorts las cuales son Turks and Caicos, Negril y Ocho Rios y se crearon page objects y specs para las páginas webs principales de Beaches y Sandals.

Durante la semana 16 se realizaron se solucionaron ciertos bugs en los specs de Sandals Butler y en Beaches Concierge y Premium Liquors por que se realizaron cambios en los selectores de estas páginas durante las promociones de enero. Se agrego la función de generar reportes en el testrunner de WebDriverIO una vez los specs hayan finalizado (Ilustración 58).

```
onComplete: async function(exitCode, config, capabilities) {
  let capability = capabilities[0];
  if (!capability.macawAPI){
    return;
  }
  let baseURL = capability.macawAPI;
  let jwt = capability.macawJWT;
  let reportsPath = filepath.join(process.env["MACAW_SRC"], ".macaw", "reports", capability.runID, "mochareport.json")
  // POST the report to the API
  let uploadedMediaList = await apiOper.uploadMedia(
    jwt,
    baseURL+"/reports/upload",
    {
      appName: capability.appName,
      filepath: fileName,
      data: { fileName: fs.createReadStream(reportsPath) },
    }
  );
  let reportData = JSON.parse(fs.readFileSync(reportsPath));
  let media = uploadedMediaList[0];
  await apiOper.updateMediaInDB(
    jwt,
    baseURL+"/reports/"+capability.appName+"/"+media._id,
    {
      runID: capability.runID,
      didpass: (exitCode===0) ? true:false,
    }
  );
};
```

Ilustración 58: Función para generar reportes

Fuente: Propia

Para finalizar en los últimos dos días de la semana se realizaron cambios en los describes y its de todos los specs de Sandals y Beaches y se cambiaron los mensajes de error por unos más específicos y mejor detallados para el usuario.

Tabla 15: Actividades realizadas Semana 14

Fecha	Descripción
Miércoles 1 de enero del 2020	Vacaciones por celebración del Año Nuevo.
Jueves 2 de enero del 2020	Se crearon nuevos page objects y specs para las páginas web de Beaches Rewards, Beaches Sandals Foundation, Our Brands y Eco Friendly
Viernes 3 de enero del 2020	Se crearon nuevos page objects y specs para las páginas de Sandals About, Groups, Rewards Program, Sandals Foundation y Eco Friendly.
Lunes 6 de enero del 2020	Se crearon nuevos page objects y specs para las páginas de Sandals What's New, Compare Resorts y Vacation Planner.
Martes 7 de enero del 2020	Se realizaron algunos cambios en los selectores de los page objects de Sandals Gourmet Dining y Entertainment debido a que estas páginas web fueron actualizadas por el equipo de Front-End.
Miércoles 8 de enero del 2020	El equipo de Front-End realizo algunos cambios en las páginas de Sandals Weddings Planner, Honeymoon y Retie The Knot por lo que se actualizaron algunos selectores en los page objects de estas páginas.
Jueves 9 de enero del 2020	Se agrego la función de tomar screenshots cuando uno de los specs falla además se agregó la función de detectar cuando una página falla por un 404 o un 500 ya que durante esta semana mucho de las pruebas fallaban por estos errores.
Viernes 10 de enero del 2020	El equipo de Front-End realizo algunos cambios en las páginas de Sandals Scuba Diving y Water Sports por lo que se actualizaron algunos selectores en los page objects de estas páginas.

Fuente: Propia

Tabla 16: Actividades realizadas Semana 15

Fecha	Descripción
Lunes 13 de enero del 2020	Se crearon nuevos page objects y specs para las nuevas páginas de Sandals Weddings las cuales son Lest Us Inspire You, Location y Test Drive Your Wedding.
Martes 14 de enero del 2020	Se realizo una reunión para determinar que se realizarán specs y page objects para las páginas que se encuentran en Sandals Resorts, Sandals Our Islands y Beaches Resorts.
Miércoles 15 de enero del 2020	Se crearon nuevos page objects y specs para las páginas de Sandals Our Islands Jamaica, Saint Lucia y Bahamas.
Jueves 16 de enero del 2020	Se crearon nuevos page objects y specs para las páginas de Sandals Our Islands Barbados, Antigua y Granada. Se realizo una reunión en la que se determinó no se crearan specs ni page object para las páginas de Sandals Resorts por el momento.
Viernes 17 de enero del 2020	Se crearon nuevos page objects y specs para las páginas de Beaches Resorts Turks and Caicos, Negril y Ocho Rios. Además, se crearon page objects y specs para las páginas webs principales de Beaches y Sandals.

Fuente: Propia

Tabla 17: Actividades realizadas Semana 16

Fecha	Descripción
Lunes 20 de enero del 2020	Se solucionaron ciertos bugs en los specs de Sandals Butler y en Beaches Concierge y Premium Liquors ya que estos fallaban por que se realizaron cambios en los selectores de estas páginas.
Martes 21 de enero del 2020	Se solucionaron problemas con el spec de página de Beaches Sesame Street ya que por algunos efectos de CSS en las imágenes están tardaban demasiado en cargar o eran invisibles para el WDIO testrunner.
Miércoles 22 de enero del 2020	Se agrego la opción de generar reportes una vez las pruebas hayan finalizado y se subió el proyecto al servidor para luego probarlo en Rundeck, estos reportes son generados por la librería de Mochawesome y una librería interna de WebdriverIO llamada specs (Ilustración 59 y 60).
Jueves 23 de enero del 2020	Se realizaron cambios en los describes y its de todos los specs de Sandals y Beaches.
Viernes 24 de enero del 2020	Se cambiaron los mensajes de error por unos más específicos y mejor detallados para el usuario. Se termino el desarrollo del proyecto.

Fuente: Propia

RUNDECK #177,700
Beaches Contents

Succeeded after 4m 19s at 1/14 4pm started at 1/14 4pm by you 0.04:18

Complete: 100% 1/1 Failed: 0 Incomplete: 0 Not Started: 0

Report Log Output Definition

View Options Text HTML Download

```

15:53:31 vanpspecsapi 1. Script -----
15:53:31 [chrome #24-0] Session ID: d658425fdf3c5488b72115e720da29536ba924c4ebf04caea8a24a503cf3da64
15:53:31 [chrome #24-0] Spec: /webroot/qa-contents/src/specs/beaches-content/inclusions_liv_plus_content.spec.js
15:53:31 [chrome #24-0] Running: chrome
15:53:31 [chrome #24-0]
15:53:31 [chrome #24-0] Should check if the Liv Plus page charge well.
15:53:31 [chrome #24-0] ✓ Should open Inclusions Liv Plus page
15:53:31 [chrome #24-0] ✓ Should check if the images in Liv Plus page are visible
15:53:31 [chrome #24-0] ✓ Should check if the Liv Plus page links redirects as expected.
15:53:31 [chrome #24-0]
15:53:31 [chrome #24-0] 3 passing (20s)
15:53:31 [chrome #24-0]
15:53:31 -----
15:53:33 [chrome #46-0] Session ID: d658425fdf3c5488b72115e720da2953e66b8a2af3c177ca0bc2e7ccba87f319
15:53:33 [chrome #46-0] Spec: /webroot/qa-contents/src/specs/beaches-content/weddings_photo_video_content.spec.js
15:53:33 [chrome #46-0] Running: chrome
15:53:33 [chrome #46-0]

```

Ilustración 59: Reportes generado por la librería interna de WebdriverIO en Rundeck.

Fuente: Propia

```

08:36:49 [chrome #19-0] Status: < 500,java.lang.NullPointerException! >: expected false to be true
08:36:49 [chrome #19-0] at Context.<anonymous> (/webroot/qa-contents/specs/beaches-content/tours_content.spec.js:77:9)
08:36:49 [chrome #19-0] at new Promise (<anonymous>)
08:36:49 [chrome #19-0] at new F (/webroot/qa-contents/node_modules/core-js/library/modules/_export.js:36:28)
08:36:49 [chrome #19-0]
-----
08:36:49 Number of specs: 55
-----
08:36:49 184 passing (302.60s)
08:36:49 4 skipped
08:36:49 1 failing
08:36:49 1) should check if the Tours page charge well. Should check if the Resorts Tours page links redirects as expected.:
08:36:49 This links has problems.
08:36:49 Links: < 0. https://www.beaches.com/tours/details/?eventId=8952&groupId=4&rstCode=B80&resortName=Beaches%20cho%20R1os,1.
08:36:49 https://www.beaches.com/tours/details/?eventId=5925&groupId=21&rstCode=B80&resortName=Beaches%20cho%20R1os >
08:36:49 Status: < 500,java.lang.NullPointerException! >: expected false to be true
08:36:49 running chrome
08:36:49 AssertionError: This links has problems.
08:36:49 Links: < 0. https://www.beaches.com/tours/details/?eventId=8952&groupId=4&rstCode=B80&resortName=Beaches%20cho%20R1os,1.
08:36:49 https://www.beaches.com/tours/details/?eventId=5925&groupId=21&rstCode=B80&resortName=Beaches%20cho%20R1os >
08:36:49 Status: < 500,java.lang.NullPointerException! >: expected false to be true
08:36:49 at Context.<anonymous> (/webroot/qa-contents/specs/beaches-content/tours_content.spec.js:77:9)
08:36:49 at new Promise (<anonymous>)
08:36:49 at new F (/webroot/qa-contents/node_modules/core-js/library/modules/_export.js:36:28)
08:36:49
08:36:49 wrote json report to [/webroot/qa-contents/.macaw/reports/Q6FBINAAA90KPLG2SPJLPFV].
08:36:49 2019/10/30 10:36:49 Command failed to run with exit status 1
08:36:49 Tests : 189
08:36:49 PassPercent : 97
08:36:49 Duration : 302552
08:36:49 Passes : 184
08:36:49 Skipped : 0
08:36:49 Failures : 1

```

Ilustración 60: Resultado final de las pruebas en Rundeck generado por la librería interna de WebdriverIO.

Fuente: Propia

4.2.2 QA QOBE

El proyecto de QA QOBE ya estaba finalizado cuando se me asignó, pero debido a que otros proyectos que el equipo de QA tenía en desarrollo considero dejar este en pausa. Mi labor en este proyecto fue de mantenimiento en casi su totalidad junto a otro miembro del equipo, se realizaron solo labores mantenimiento debido a que la aplicación de OBE (Online Booking Engine) a la cual aplica este proyecto enfoca sus pruebas (tests o specs) cambio en gran medida a lo largo de su desarrollo por ende varias pruebas quedaron obsoletas y con varios bugs debido a que se removieron secciones o se cambiaron los selectores CSS de la aplicación de OBE.

4.2.2.1 Definición del Problema

Al iniciar la creación del proyecto de OBE se observó que la mayoría de las pruebas manuales que realizaba el equipo de Manual QA de India en este proyecto eran muy parecidas o seguían un cierto patrón muy marcado por lo que al equipo de QA de San Services propuso que había material suficiente para crear test automatizados y liberar a los manuales testers de este trabajo repetitivo.

4.2.2.2 Descripción General del Proyecto


El proyecto de QA QOBE consiste en verificar la estructura y el funcionamiento de las aplicaciones de Sandals OBE, Beaches OBE y Grand Pineapple OBE. En palabras generales cuando se refiera a verificación de la estructura de la página se refiere a si los precios presentes estén correctos, la visibilidad y disponibilidad de las imágenes, estado de los enlaces y el redireccionamiento a otras páginas mediante los botones, funcionamiento de popup, el funcionamiento correcto de los formularios entre otras cosas.


El proyecto verifica el estado de las cinco páginas web denominados como steps que contienen las tres aplicaciones de OBE, estas páginas web son Vacation (Step 1), Room (Step 2), Flights (Step 3), Guests (Step 4) y Payment (Step 5).


Vacation — Room — Flights — Guests — Payment

More Quality Inclusions than any other resort on the planet
[> View Inclusions](#)


Select Vacation Dates:


CHECK IN: 

CHECK OUT: 

Select Sandals Resort: 

Include Roundtrip Flights? No, thank you Yes, please

Flights departing from: 

Search flights by: 

Results will be displayed in the "Flights" section.

Ilustración 61: Pagina de OBE Vacation

Fuente: Propia

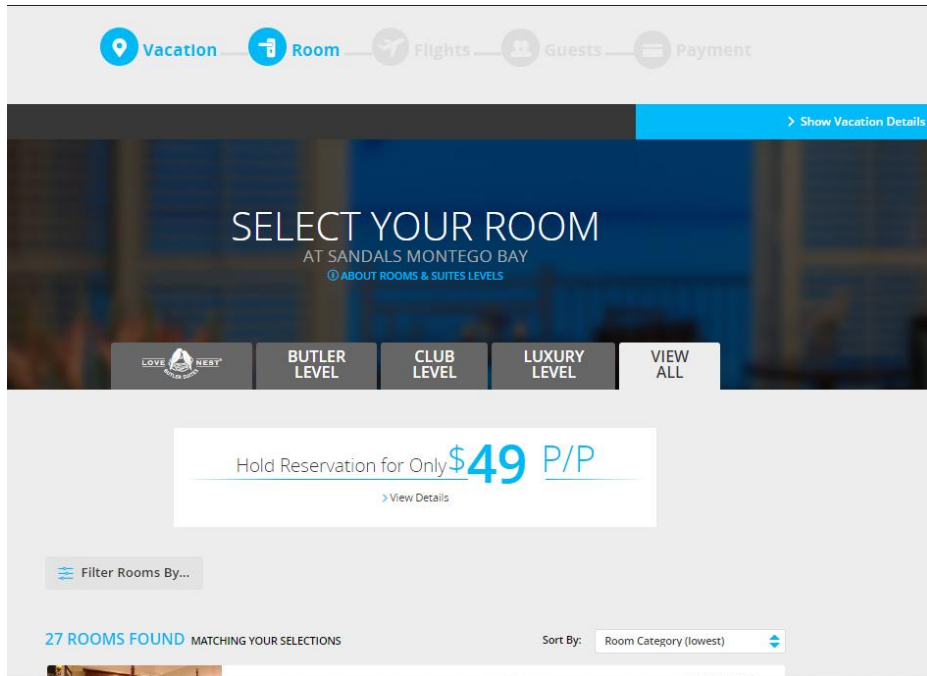


Ilustración 62: Pagina de OBE Room

Fuente: Propia

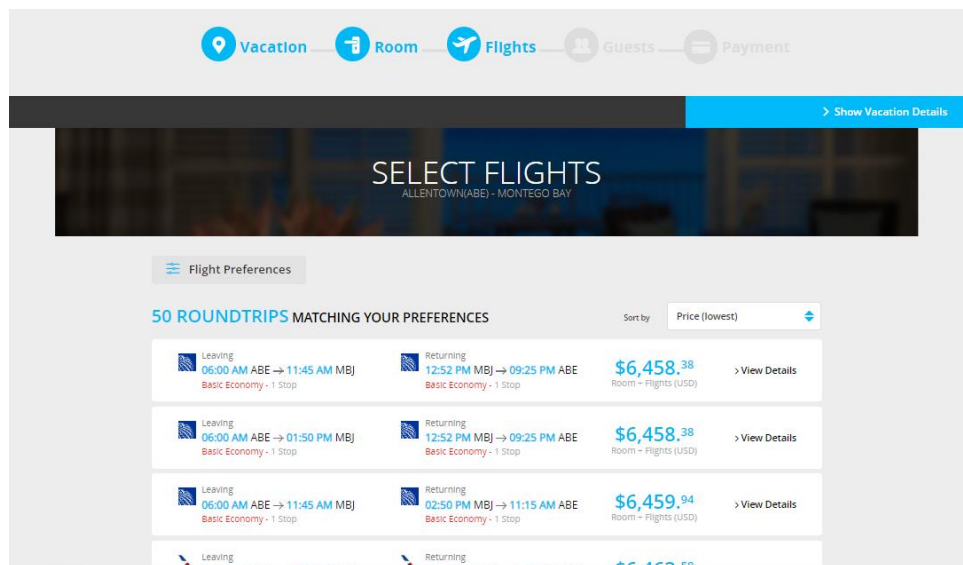


Ilustración 63: Pagina de OBE Flights

Fuente: Propia

Vacation Room Flights **Guests** Payment

> Show Vacation Details

PRIMARY GUEST

MUST BE 18 YEARS OLD OR OLDER

Please be advised that names entered here must match your proof of citizenship at time of travel. Change or cancellation penalties will apply for misspelled or incorrect names. Please DO NOT enter P.O. BOXES as we do not send travel documents to them.

* REQUIRED FIELD

Title *

Gender *

First Name *

Middle Name

Last Name *

Date of Birth *

Ilustración 64: Pagina de OBE Guests

Fuente: Propia

Vacation Room Flights **Guests** **Payment**

PLEASE REVIEW

✓ You will be able to add special requests and comments once the booking is completed.

Travel Information

Resort Selected: Sandals Montego Bay
Location: Montego Bay, Jamaica

Check-in: Monday, April 6th, 2020
Check-out: Tuesday, April 14th, 2020

Length Of Stay: 8 Nights

Room Selection

Room Category: Caribbean Deluxe - DL
Room Features: In-Room Local Beer & Wine, Free Wifi

Ilustración 65: Pagina de OBE Payment

Fuente: Propia

Debido a la magnitud de cambios que había por realizar en este proyecto se fue asignado a mi persona a otro integrante del equipo por lo tanto se creó un tablero nuevo de Trello en el cual se asignarían las tareas a realizar (Ilustración 66).

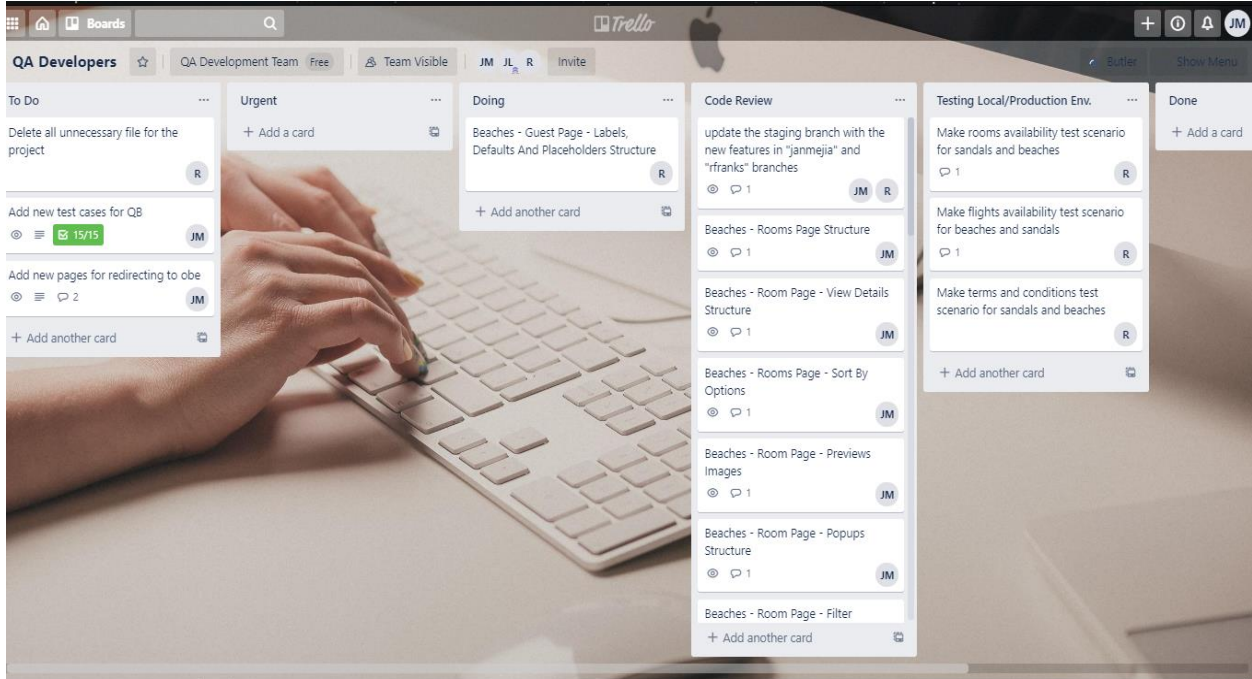


Ilustración 66: Tablero de tareas para el proyecto de OBE

Fuente: Propia

4.2.2.3 Bitácora de Actividades Realizadas

La bitácora del trabajo realizado en este proyecto se presentará solo con una bitácora diaria de las actividades realizadas por semanas del mes. Ya que en este proyecto lo que se realizó fue puro mantenimiento del código, no se explicará a mucho detalle ya que el mostrar código o explicar el funcionamiento de los specs mostraría debilidades del proyecto al cual se le realizan las pruebas automatizadas.

Tabla 18: Actividades realizadas Semana 17

Fecha	Descripción
Lunes 27 de enero del 2020	Se realizo una reunión para asignarme al proyecto de QA OBE ya que el proyecto estaba algo desactualizado con los nuevos cambios que se habían realizado por ende fui añadido a este proyecto junto a otro miembro del equipo para dar mantenimiento a la aplicación.
Martes 28 de enero del 2020	Se realizaron pruebas en todos los specs tanto de Sandals y Beaches para de esto modo enumerar de forma rápida cuales eran las pruebas que retornaban errores por cambio de selectores o por que la paginas web ya no contaban con ciertas secciones que los specs revisaban en un inicio.
Miércoles 29 de enero del 2020	Se realizaron pruebas en todos los specs de Grand Pineapple para realizar lo mismo del martes 28. Se agregaron a Trello todos los specs que fallaron en OBE y se me dio la tarea de solucionar algunos de estos.
Jueves 30 de enero del 2020	Se le dio mantenimiento a los specs encargado de verificar el tamaño máximo de los inputs de Sandals OBE Payment y al encargado de verificar los casos que despliegan errores, ya que varias secciones que revisaban estos specs dejaron de existir por lo que el código se volvió obsoleto, además se agregaron nuevas funciones para adaptarlo a la nueva estructura de la página web de Sandals OBE Payment.
Viernes 31 de enero del 2020	Se le dio mantenimiento a los specs encargados de verificar los casos que despliegan errores al momento de ingresar datos inválidos en la sección de tarjetas de crédito y al spec encargado de verificar el funcionamiento de selección de país de Sandals OBE Payment a todo esto se le dio mantenimiento por la misma razón del jueves que varias secciones de la página web dejaron de existir.

Fuente: Propia

Tabla 19: Actividades realizadas Semana 18

Fecha	Descripción
Lunes 3 de febrero del 2020	Se le dio mantenimiento al spec encargado de verificar que la información seleccionada en las secciones de Sandals OBE Vacation y Rooms concuerden con la información desplegada en Sandals OBE Flights.
Martes 4 de febrero del 2020	Se le dio mantenimiento al spec encargado de verificar que la información seleccionada en las secciones de Sandals OBE Vacation, Rooms y Flights concuerden con la información desplegada en Sandals OBE Guests, a este spec se le aplico el mismo mantenimiento que al spec realizado el día de ayer.
Miércoles 5 de febrero del 2020	Se le dio mantenimiento a los specs encargados de revisar las secciones de la página de Sandals OBE Rooms debido a que en la página de Sandals OBE Vacation se generaron cambios en la selección del resort, por ende, todos los specs que acceden a la página de Rooms mediante Vacation se les dio mantenimiento.
Jueves 6 de febrero del 2020	Se le dio mantenimiento a los tres specs de Sandals OBE Payment que verifican el proceso de layaway, debido a que varios selectores y funciones quedaron obsoletas.
Viernes 7 de febrero del 2020	Se le dio mantenimiento a dos specs de Sandals OBE Payment que verifican la entrada de información errónea y valida en la sección de ingreso de tarjeta de crédito y se le dio mantenimiento al spec encargado de verificar el funcionamiento del verificador de la ley de Luhn.

Fuente: Propia

Tabla 20: Actividades realizadas Semana 19

Fecha	Descripción
Lunes 10 de febrero del 2020	Se le dio mantenimiento al spec encargado de verificar el proceso de modificación de información en la página de Sandals OBE Payment y también se le dio mantenimiento al spec que verifica las opciones de pago en esta misma página.
Martes 11 de febrero del 2020	Se le dio mantenimiento al spec encargado de verificar que la información seleccionada en secciones anteriores concuerde con la mostrada en la página de Sandals OBE Payment.
Miércoles 12 de febrero del 2020	Se subieron todos los cambios realizados hasta hoy al repositorio de QA QOBE. Se realizó una migración para el spec que verifica la estructura de completa de la página de Sandals OBE Payment y se le realizo mantenimiento al spec que verifica el popup de términos y condiciones de Payment, debido a que se le cambio el selector al botón que despliega el popup.
Jueves 13 de febrero del 2020	Se agregaron nuevas direcciones de servidores para realizar pruebas en las páginas de Sandals y Beaches OBE. Se le dio mantenimiento al spec que verifica el popup de términos y condiciones de Beaches OBE Payment, debido a que se le cambio el selector al botón que despliega el popup.
Viernes 14 de febrero del 2020	Se cambiaron y agregaron nuevos selectores al Brand Object de Grand Pineapple y se les dio mantenimiento a los specs que verifican la disponibilidad de los vuelos en la página de Grand Pineapple OBE Flights.

Fuente: Propia

Tabla 21: Actividades realizadas Semana 20

Fecha	Descripción
Lunes 17 de febrero del 2020	Se realizo una reunión en la que se determinó el uso de una nueva implementación sobre el manejo de los calendarios presentes en las páginas de Sandals, Beaches y Grand Pineapple.
Martes 18 de febrero del 2020	Se agregaron nuevas direcciones de servidores para realizar pruebas en la página de Grand Pineapple OBE. Se le dio mantenimiento a los specs que seleccionan fechas en el calendario de Beaches OBE Vacation debido a que se cambiaria la implementación del manejo de calendarios actual por la nueva implementación discutida en la reunión del día de ayer.
Miércoles 19 de febrero del 2020	Se le dio mantenimiento a los specs que seleccionan fechas en el calendario de Sandals OBE Vacation debido a que se cambiaria la implementación del manejo de calendarios actual por una nueva implementación.
Jueves 20 de febrero del 2020	Se continuo con el mantenimiento a los specs que seleccionan fechas en el calendario de Sandals OBE Vacation por el cambio de implementación del manejo de los calendarios.
Viernes 21 de febrero del 2020	Se le dio mantenimiento a los specs que seleccionan fechas en el calendario de Grand Pineapple OBE Vacation debido a que se cambiaria la implementación del manejo de calendarios actual por una nueva implementación.

Fuente: Propia

Tabla 22: Actividades realizadas Semana 21

Fecha	Descripción
Lunes 24 de febrero del 2020	Se le dio mantenimiento a los specs que verifican la estructura de las páginas de Sandals OBE Rooms y Beaches OBE Rooms.
Martes 25 de febrero del 2020	Se realizaron migraciones para los tres specs de Beaches que se encargan de generar un booking completo y se añadieron nuevos selectores a los Brand Object de Sandals y Beaches.
Miércoles 26 de febrero del 2020	Se realizaron migraciones para los tres specs de Grand Pineapple que se encargan de generar un booking completo y realizaron también migraciones para specs de Sandals que también se encargan de lo mismo.
Jueves 27 de febrero del 2020	Se dio mantenimiento al step object que se encarga de realizar las acciones de selección de fechas en los calendarios y se solucionó un bug en el spec que se encarga de revisar si hay vuelos disponibles en la página de Sandals OBE Flights.
Viernes 28 de febrero del 2020	Se realizo una reunión para determinar si los specs que se encargan de verificar el funcionamiento de todos los filtros en la página de OBE Rooms pasarían a ser specs individuales para probar el funcionamiento de cada filtro. Lo cual se aprobó por lo tanto se procedió a realizar los nuevos specs para la aplicación de cada uno de los filtros.

Fuente: Propia

Tabla 23: Actividades realizadas Semana 22

Fecha	Descripción
Lunes 2 de marzo del 2020	Se solucionaron bugs generados por el cambio de selectores en los specs para Sandals y Beaches OBE Flights y Payment
Martes 3 de marzo del 2020	Se les dio mantenimiento a los specs que se encargan de verificar la estructura de la página de Beaches OBE Guests y se actualizaron los selectores en el page object de Beaches OBE Guests.
Miércoles 4 de marzo del 2020	Se les dio mantenimiento a los specs de Sandals OBE Flights ya que varios de estos specs quedaron obsoletos por cambios drásticos en ciertas secciones de esta página.
Jueves 5 de marzo del 2020	Se continuo con el mantenimiento a los specs de Sandals OBE Flights.
Viernes 6 de marzo del 2020	Se termino el mantenimiento de los specs de Sandals OBE Flights y se agregó un nuevo spec que se encarga de verificar los errores mostrados en caso de ingresar datos inválidos en la página de Sandals OBE Vacation y Beaches OBE Vacation.

Fuente: Propia

Tabla 24: Actividades realizadas Semana 23

Fecha	Descripción
Lunes 9 de marzo del 2020	Se realizo una migración al spec que se encarga de verificar los términos de condiciones en Sandals OBE Payment, Beaches OBE Payment y Grand Pineapple Payment.
Martes 10 de marzo del 2020	Se les dio mantenimiento a varios specs que se encargan de verificar la estructura de las páginas de Beaches OBE Flights y Grand Pineapple Flights.
Miércoles 11 de marzo del 2020	Se continuo con el mantenimiento de los specs de Grand Pineapple Flights.
Jueves 12 de marzo del 2020	Se cambiaron varios selectores en los page objects de OBE Vacation y Guests tanto de Sandals, Beaches y Grand Pineapple
Viernes 13 de marzo del 2020	Se removieron varios its y describes de los specs de Beaches OBE Payment, Guests, Rooms y Vacation debido a que estos verificaban áreas no importantes en las páginas de Beaches OBE.

Fuente: Propia

V. CONCLUSIONES

- La fácil integración a los dos proyectos fue debido al proceso de capacitación que se realizó al inicio de la práctica profesional ya que esto permitió conocer las tecnologías, estructura y metodología utilizada en los proyectos.
- El proceso de mantenimiento en uno de los proyecto asignados no se finalizó debido a que eran demasiadas pruebas automatizadas que estaban desactualizadas y el periodo de practica ya venía llegando a su fin, pero se le brindo mantenimiento a una gran parte de ellos.
- Se finalizo el desarrollo de uno de los proyecto asignados durante el periodo de práctica, manteniendo las buenas prácticas de código limpio, las mismas herramientas, tecnologías y la estructura implementada en los proyectos anteriormente desarrollados, por el departamento.

VI. RECOMENDACIONES

- Se recomienda al departamento de Quality Assurance, que se actualicen ciertas tecnologías como ser WebdriverIO debido a que la nueva versión tiene una mejor compatibilidad con ReactJS que la versión usada actualmente.
- Se recomienda al departamento de Quality Assurance, que cambie la estructura general de los proyecto por una más reducida, debido a que hay varias secciones de la estructura que hacen uso del polimorfismo de manera innecesaria y bien podrían ser removidas, además que en proyectos muy grandes se dificultan en gran medida el darles mantenimiento a las pruebas automatizadas.
- Se recomienda al departamento de desarrollo web en general, que realice reuniones semanales con el departamento de Quality Assurance para que de este modo se evite darle mantenimiento una gran cantidad de pruebas automatizadas a la vez, ya que el mantenimiento que se le realizo al último proyecto asignado se realizó por falta de comunicación.

VII. BIBLIOGRAFÍAS

adamgruber/mochawesome: A Gorgeous HTML/CSS Reporter for Mocha.js. (s/f). Recuperado el 1 de marzo de 2020, de <https://github.com/adamgruber/mochawesome>

Bahit, E. (2011). *Scrum y eXtrem Programming para programadores.*

Chacon, S., & Straub, B. (2009). *Pro Git.*

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). *Scrum Prime.*

Eguiluz, J. (2008). *Introduccion a JavaScript.* 134.

Expect / Should—Chai. (s/f). Recuperado el 9 de febrero de 2020, de <https://www.chaijs.com/api/bdd/>

Getting Started · WebdriverIO. (s/f). Recuperado el 9 de febrero de 2020, de <https://webdriver.io/index.html>

JavaScript | MDN. (s/f). Recuperado el 1 de marzo de 2020, de <https://developer.mozilla.org/es/docs/Web/JavaScript>

Kaoru, I. (1989). *Introduccion al Control de Calidad.*
https://drive.google.com/file/d/1pdBFPbKGOASwJgKtQ6EJPUx7D0PcPaUJ/view?usp=drive_open&usp=embed_facebook

Mocha—The fun, simple, flexible JavaScript test framework. (s/f). Recuperado el 9 de febrero de 2020, de <https://mochajs.org/>

Node.js. (s/f). *Acerca.* Node.js. Recuperado el 5 de mayo de 2020, de <https://nodejs.org/es/about/>

Red Hat. (s/f). *¿Qué es la automatización de la TI?* Recuperado el 27 de febrero de 2020, de <https://www.redhat.com/es/topics/automation/whats-it-automation>

Rundeck Documentation. (s/f). Recuperado el 9 de febrero de 2020, de <https://docs.rundeck.com/>

Selenium UI. (s/f). Recuperado el 9 de febrero de 2020, de <https://aerokube.com/selenium-ui/latest/>

Simpson, K. (2015). *You Dont Know JS Up and Going* (1a ed.).

Sistema de control de calidad de software—EcuRed. (s/f). Recuperado el 1 de marzo de 2020, de https://www.ecured.cu/Sistema_de_control_de_calidad_de_software

Studio, F. (2019, enero 29). *What Is QA And Why Quality Assurance Is Important in Web Development Process*. Medium. <https://medium.com/@f5studio/what-is-qa-and-why-quality-assurance-is-important-in-web-development-process-f17ae9c59de7>

Usage Guide · Babel. (s/f). Recuperado el 9 de febrero de 2020, de <https://babeljs.io/>