



**Certified Tech  
Developer**

The Ultimate Degree

**DigitalHouse**>  
Coding School

## Reporte Final de Testing

# Wesped Grupo 9

### Integrantes:

- Maira Alejandra Marin Morales
- Andrés Felipe Fajardo Parra
- Dina Luz Perez Montes
- R. Indira Valentina Réquiz Molina
- Juan Ignacio Basualdo
- Pablo Alexander Alarcón Martinez
- Romina Vilatta Montenegro



# Certified Tech Developer

The Ultimate Degree

DigitalHouse>  
Coding School

Contenido	Pág.
Introducción	3
Relación de Issues	3
Resumen de las actividades	6
Sprint 1	6
Issue #12 Planificación y ejecución de los tests	6
Issue #13 Testear la API	8
Sprint 2	8
Issue #41 Testeo Automatizado (JEST)	9
Issue #42 Testeo Automatizado (Selenium)	10
Issue #43 Testeo Automatizado (Postman)	10
Issue #44 Testeo Exploratorio - Manual - Estático	10
Sprint 3	12
Issue # 66 Implementar tests manuales	12
Issue # 67 Implementar tests Automatizados	12
Sprint 4	14
Issue # 74 Implementar testing Automatizado	14
Issue # 75 Implementar testing manual	14
Issue # 76 Confeccionar el reporte final de testing	14
Informe de defectos	15



## Introducción

El siguiente documento corresponde al informe final de testing del equipo 9 y su proyecto integrador para el primer año de la carrera Certified Tech Developer de Digital House. Este tiene como objetivo dar a conocer los resultados de Wespel, nuestro proyecto, un sitio para reservas de hoteles.

El documento contiene la información relacionada al avance del desarrollo de un MVP, con la metodología SCRUM, que se planteó en 4 sprints, correspondiente al testing.

El testing se basó en las issues, las cuales fueron planteadas como requerimientos técnicos para llegar al objetivo esperado. Para ello se implementaron casos de prueba, cuya ejecución se realizó con técnicas manuales y automatizadas; lo que permite al equipo 9 recopilar las evidencias sobre el avance del proyecto, las que serán expuestas a continuación.

En el proceso del proyecto integrador fue solicitada la realización de varias issues enfocadas a testing. Les presentamos dichos requisitos y posteriormente mostramos la evaluación de su estado.

## Relación de Issues

Nº de Issue	Sprint	Contenido	Avance
12	1	Planificación y ejecución de los tests Planear casos de test basados en las historias de usuarios que existen en el sprint, y luego ejecutarlos.  Realizar también una prueba exploratoria sobre lo que se desarrolló durante el Sprint.	Finalizado
13	1	Testear la API	Finalizado



Validar el funcionamiento de la API Rest creada por el Back End durante el Sprint.

Utilicen Postman para las pruebas y creen los scripts de prueba necesarios.

41	2	Testeo Automatizado (JEST) Creación de pruebas automatizadas	Finalizado
----	---	---	------------

Utilizando el Framework JEST, crear los scripts de prueba automatizados necesarios para cubrir lo desarrollado durante los sprints I y II. Subir evidencia de la ejecución de los test a la tarea de GitLab (captura de pantalla o archivo JSON).

Crear test unitarios y definir el porcentaje de cobertura que quiere alcanzar el equipo (como mínimo un 40 %).

42	2	Testeo Automatizado (Selenium) Opcional: Crear test funcionales utilizando Selenium Web driver.	Finalizado
----	---	--	------------

Opcional: Crear script automático utilizando Selenium IDE.

43	2	Testeo Automatizado (Postman) Creación de pruebas automatizadas	Finalizado
----	---	--	------------

Desarrollar los scripts de pruebas automatizados utilizando Postman para las API desarrolladas en este sprint. Agregar las API desarrolladas en este sprint a la colección generada en el sprint I. Ejecutar los scripts con el test runner de Postman y subir a GitLab el archivo JSON con los resultados de la ejecución (debe contener los scripts de sprint I y II).

44	2	Testeo Exploratorio - Manual - Estático Testing exploratorio	Finalizado
----	---	---	------------

Realizar test exploratorio sobre lo agregado en el sprint. Entregar un documento con notas y forma de



la organización del test exploratorio (sesiones, tours, escenarios, workflows, etc.).

Casos de prueba manuales

Agregar a la planilla de casos de prueba sobre las funcionalidades del sprint II.

Realizar mantenimiento de los casos de prueba manuales generados en el sprint I, en caso de ser necesario.

Clasificar los casos de prueba en suite de smoke y regression test.

Generar la suite de prueba con los casos que se puedan ejecutar, luego ejecutar esta suite de prueba.

Subir a GitLab la planilla de casos de prueba adjunta o el link al archivo.

Testing estático

Revisar cada una de las historias de usuario del sprint II y reportar defectos en la planilla correspondiente (por ejemplo, falta de definición de algún componente).

Subir a GitLab la planilla de defectos adjunta o el link al archivo.

66	3	Implementar tests manuales Realizar pruebas exploratorias en las páginas de reserva y de confirmación de reserva. Verificar y validar las funcionalidades y la interfaz de usuario. Agregar test cases manuales sobre las funcionalidades a entregar en este sprint y ejecutar un ciclo de prueba seleccionando tests cases necesarios, teniendo en cuenta de no duplicar esfuerzo (evitando de ejecutar los que ya se encuentran automatizados).	Finalizado
67	3	Implementar tests automatizados Añadir las pruebas necesarias para los endpoints de productos, reservas y usuarios a la colección creada previamente en Postman (scripts de pruebas	Finalizado



		<p>automatizados).</p> <p>Añadir / Crear un script relacionado al proceso de reserva en las pruebas de regresión automatizadas creadas con Selenium IDE (o Selenium Web Driver).</p> <p>Opcional: Crear y ejecutar las pruebas unitarias (unit testing) de las páginas de reserva y confirmación de reserva a través de JEST framework</p>	
74	4	<p>Implementar testing automatizado</p> <p>Crear y ejecutar test unitarios utilizando Selenium sobre las funcionalidades de la página de mis reservas (las realizadas por usuario) y sobre la página formulario de producto (la que se ingresa con rol administrador).</p> <p>Agregar a la colección de Postman script automatizados para probar los métodos agregados en este sprint.</p>	Finalizado
75	4	<p>Implementar testing manual</p> <p>Realizar y registrar el testing exploratorio sobre las funcionalidades de la página de mis reservas y sobre la página formulario de producto.</p> <p>Agregar casos de prueba Manuales para ambas páginas y clasificar los mismos en Smoke y Regression Test.</p>	Finalizado
76	4	<p>Confeccionar el reporte final de testing</p> <p>Ejecutar los casos de prueba que pertenecen a la suite de Regresión y generar un Reporte Final de Prueba que contenga el siguiente resumen (pueden ser gráficos o tablas agregadas en el template) :</p> <p>Cantidad de casos de prueba creados por funcionalidad/historia de usuario tanto manuales como automáticos.</p> <p>Cantidad de casos de prueba ejecutados, pasados y fallados.</p>	Finalizado



Cantidad de defectos por estado y prioridad/severidad  
Como exit criteria (criterio para finalizar el testing)  
no se debería tener defectos en estado abierto de severidad crítica y/o bloqueante.

## Resumen de Actividades

Les invitamos a conocer nuestro proyecto:

Viva la experiencia de reserva más completa [Wesped](#)

## Sprint 1

### Issue #12 Planificación y ejecución de los tests

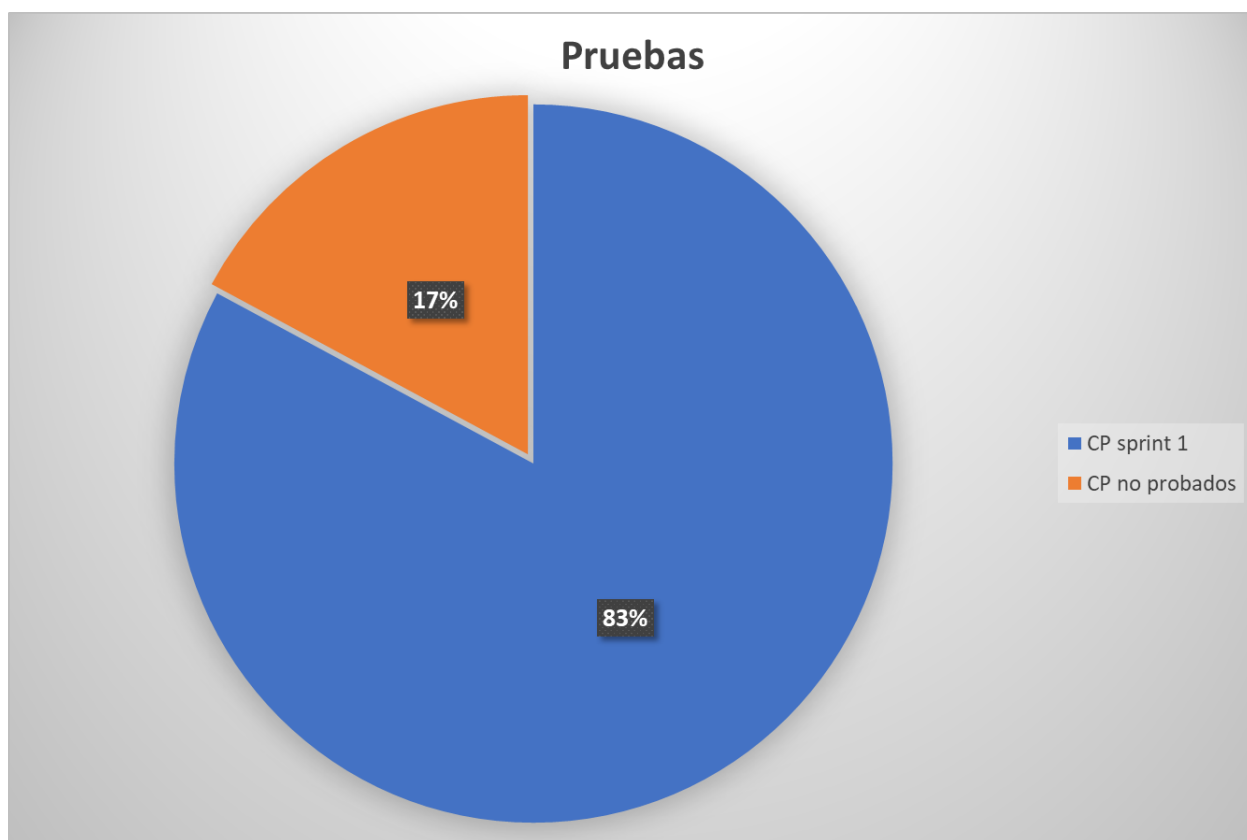
Se pide realizar la planeación de casos de prueba. Estos están basados en las historias de usuarios que existen en el sprint; y después ejecutar los test necesarios para desarrollar los respectivos casos, el equipo decide elaborar un archivo excel con distintas ventanas. En la primera se encuentran los casos de prueba, en la cual detallamos cada caso de prueba con un "Id" (CP-001), la numeración nos permite el conteo de cuántos casos llevamos. Esta hoja también cuenta con los siguientes datos:

- Fecha de creación
- Título
- Descripción
- Área de Desarrollo
- Precondición
- Pasos: contiene Acción y Resultado esperado
- Estado
- Tipo de Prueba
- Reportado por



Además de estos ítems se encuentra resaltado el sprint al cual pertenecen los casos de prueba, lo que nos da la posibilidad de tener una secuencia y orden en la ejecución de las pruebas.

Se realizó el test referente a los casos de prueba:



## Issue #13 Testear la API

En lo relacionado con el Backend, se manejaron los métodos (Get, Get por Id, Post, Delete) para realizar las pruebas de la API. Se agregó una variable para favorecer la automatización de las pruebas y, además se agregó test 200 para validar el pedido del requerimiento.

En POST y DELETE se agregaron test de validación con mensajes que corroboraron la aceptación y respuesta de las peticiones positivamente.



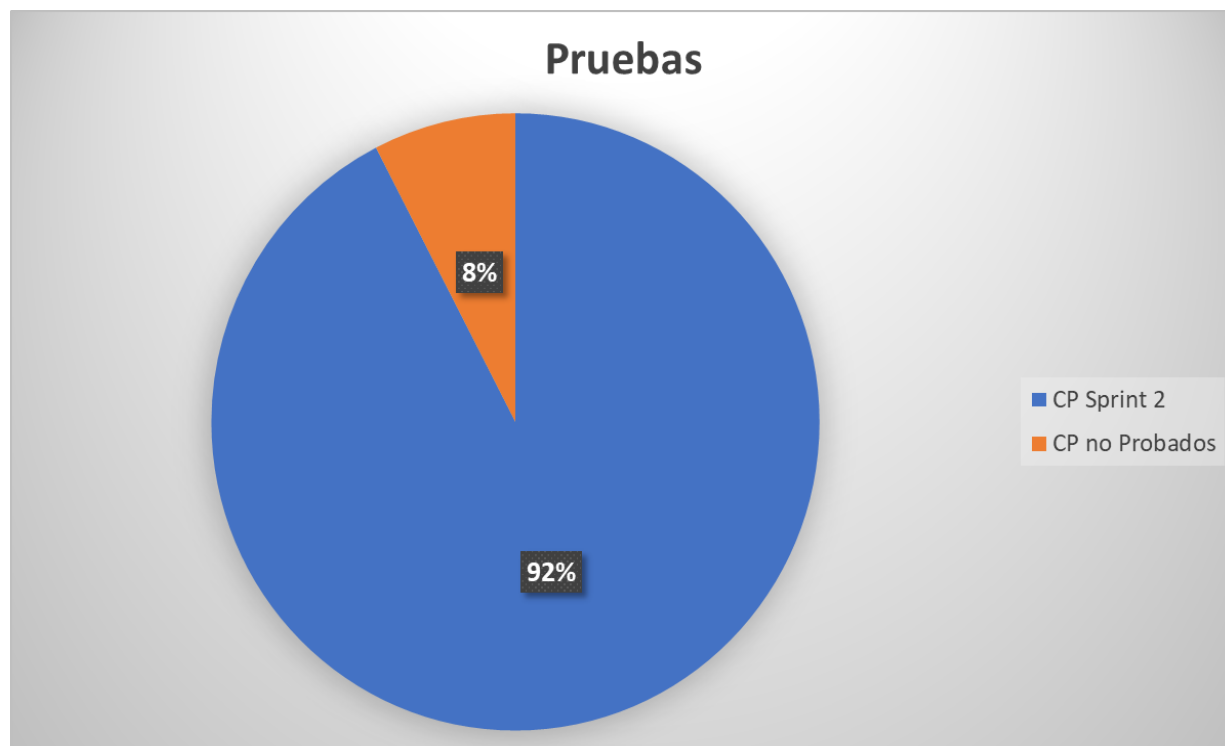


Por último se generó el archivo JSON, script, con los requerimientos efectuados en Postman, para agregarlo como soporte en la carpeta del gitlab.

## Sprint 2

En este sprint se realizaron 39 casos de prueba para todos los test concernientes al frontend y el backend. También se generaron algunas pruebas con la herramienta Jest y Postman; a su vez se agregaron las correspondientes evidencias a la carpeta de testing relacionada al sprint 2.

Como complemento se realizaron pruebas exploratorias y manuales para determinar fallos en la ejecución de la página Wesped.



### Issue #41 Testeo Automatizado (JEST)



En este punto se tuvieron dificultades, ocasionadas al desconocimiento de la herramienta. Sin embargo, se desarrollaron algunas pruebas por parte del equipo y con ello se cubrieron algunos aspectos del sprint.

## RESULTADO

```
> frontend@0.1.0 test:coverage
> jest --coverage
```

PASS

src/components/button/Button.test.js

PASS

src/components/footer/Footer.test.js

PASS

src/components/card/Card.test.js

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	100	100	100	
components/button	100	100	100	100	
Button.js	100	100	100	100	
StyledButton.js	100	100	100	100	
components/card	100	100	100	100	
Card.js	100	100	100	100	
CardStyles.js	100	100	100	100	
components/footer	100	100	100	100	
Footer.js	100	100	100	100	
FooterStyled.js	100	100	100	100	
pages/home/recommendations/recommendationsItems	100	100	100	100	
RecommendationsItemStyle.js	100	100	100	100	

Test Suites: 3 passed, 3 total

Tests: 4 passed, 4 total

Snapshots: 0 total

Time: 7.56 s

Ran all test suites.

## Issue #42 Testeo Automatizado (Selenium)

Esta issue era opcional y no se realizó en este sprint pruebas.

## Issue #43 Testeo Automatizado (Postman)

La elaboración de este sprint se basó en agregar productos, características, e imágenes. Esto llevó al desarrollo de la plataforma para llevar a la nube los componentes del proyecto, donde se realizaron Bucket S3 de AWS para contener las fotos en la nube, y una EC2 para contener los productos. Esta infraestructura contiene las imágenes, características, y demás componentes del proyecto.



# Certified Tech Developer

The Ultimate Degree

DigitalHouse>  
Coding School

## Issue #44 Testeo Exploratorio - Manual - Estático

En esta issue se realiza la exploración del producto en la mayoría de componentes para verificar la funcionalidad de la página. Durante las pruebas se encontraron inconsistencias, las cuales fueron reportadas.



# Certified Tech Developer

The Ultimate Degree

DigitalHouse >  
Coding School



Encuentra la estadía de tus sueños

SOPORTE DE PRUEBAS

Operativo/Navegador	Desktop
Realizado por:	Dina Perez
Fecha:	24/08/2022

**Título:** CPE-008 Validar login del aplicativo

**Descripción:** Validar que el usuario pueda acceder a la aplicación cuando coloca email y contraseña correcta

**Pasos para la reproducción:**

1. Abrir la aplicación WEB desde Desktop
2. Dar clic en el botón iniciar sesión
3. Colocar Email y contraseña válida
4. Dar clic en el botón ingresar

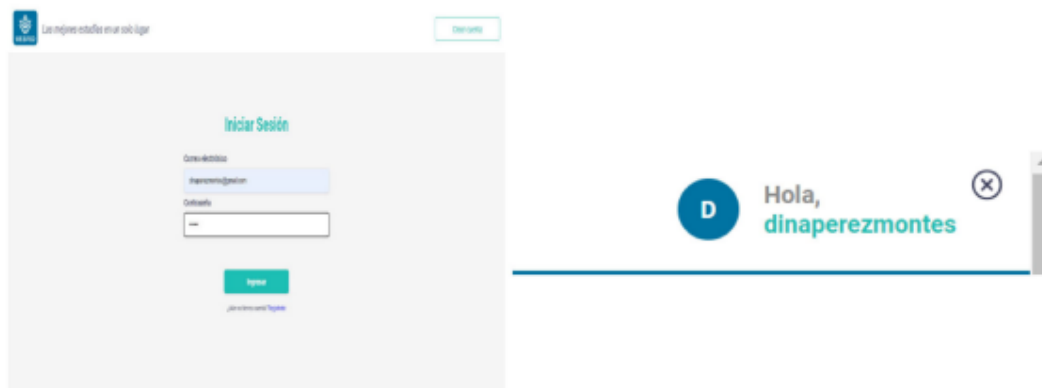
**Resultado Esperado:**

Que al colocar los datos requeridos el usuario pueda acceder a la aplicación de manera satisfactoria.

**Resultado Actual:**

El usuario puede acceder a la aplicación de manera satisfactoria siguiendo los pasos establecidos.

**Impresión de pantalla:**





## Sprint 3

Para este sprint el equipo de testing desarrolló 36 casos de prueba correspondientes a las nuevas características del proyecto, en las que se tuvieron en cuenta las políticas, el usuario, las reservas, la seguridad. Todo ello sin afectar funcionalidades previas.

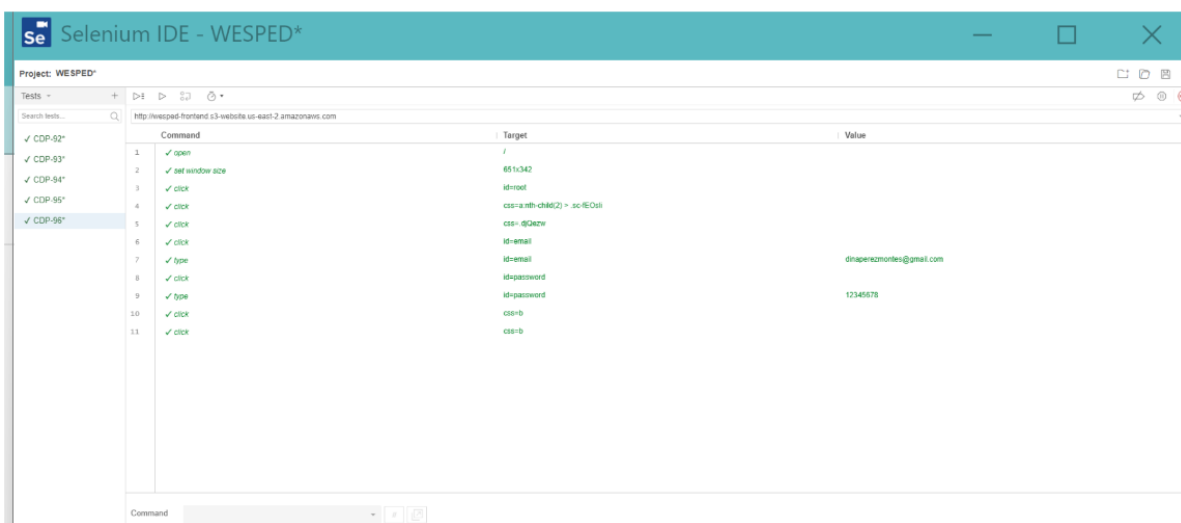
### Issue # 66 Implementar tests manuales

En esta issue tenemos la verificación de funcionalidades en la página principal, evaluar que el usuario pueda registrarse, loguearse, con su mail y contraseña válidos y, luego intentar el logueo con datos erróneos. También se prueba la funcionalidad con el usuario logueado, evaluar los productos, las fechas de reserva y sus funcionalidades.

### Issue # 67 Implementar test automatizados

Para esta issue se agregaron métodos buscar por ciudad y buscar por categoría. Entonces, se generó el crud con las respectivas peticiones para listar el producto, seleccionar por ciudad, realizar la búsqueda por categoría, loguear a los usuarios, realizar las reservas entre otras funcionalidades.

#### CDP-96 Validar la contraseña



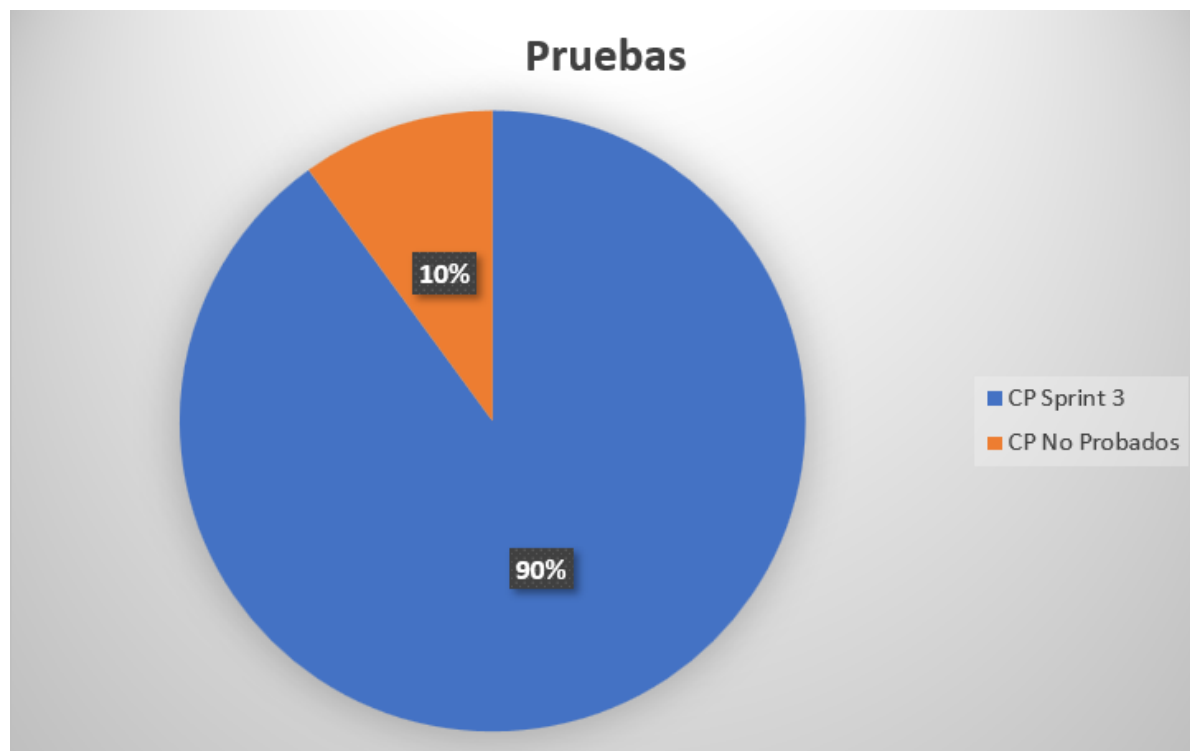


# Certified Tech Developer

The Ultimate Degree

DigitalHouse>  
Coding School

```
{
  "info": {
    "_postman_id": "87e05cc1-a51b-4f64-a75e-91a3cc4a657c",
    "name": "Proyecto Integrador",
    "description": "# Certified Tech Developer\\n\\n## Digital House\\n\\n### Proyecto Integrador - Equipo 9\\n\\n#### 2022",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
    "_exporter_id": "21444438"
  },
  "item": [
    {
      "name": "Categories",
      "item": [
        {
          "name": "GET - Categories",
          "event": [
            {
              "listen": "test",
              "script": {
                "exec": [
                  "pm.test(\"Status code is 200\", function () {\r",
                  "    pm.response.to.have.status(200);\r",
                  "});"
                ],
                "type": "text/javascript"
              }
            }
          ],
          "request": {
            "method": "GET",
            "header": [],
            "url": {
              "raw": "{{BASE_URL}}/categories/",
              "host": [
                "{{BASE_URL}}"
              ],
              "path": [
                "categories",
                ""
              ]
            }
          }
        }
      ]
    }
  ]
}
```



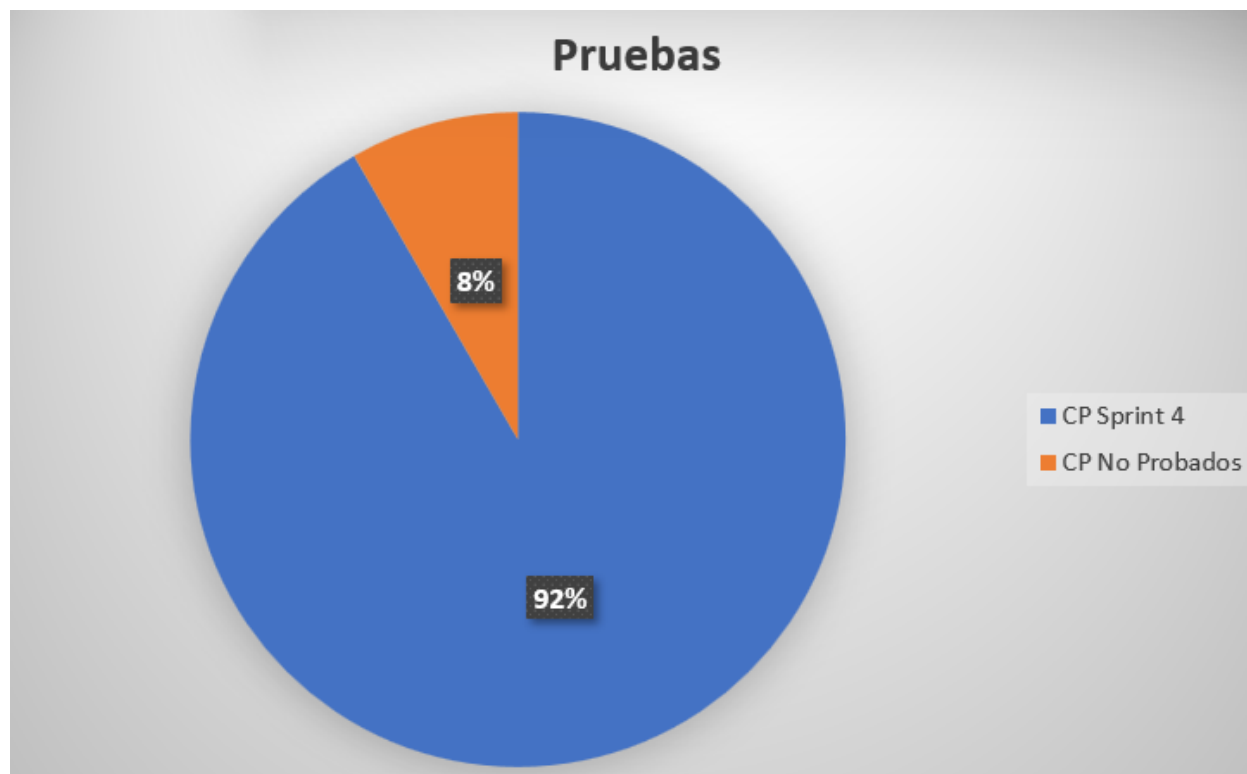
## Sprint 4

### Issue # 74 Implementar testing automatizado

Para esta issue tenemos la necesidad de loguearse como administrador y así poder evaluar las reservas con la automatización.

### Issue # 75 Implementar testing manual

En el desarrollo del sprint manejamos la evaluación de 11 casos para el testing manual; enfocados en la creación de un producto y se realizaron pruebas de regresión para la confirmación de una reserva y la validación del calendario en detalle del producto.



## Issue # 76 Confeccionar el reporte final de testing

Se desarrolla el contenido de los cuatro sprint, con la consolidación de los datos en el informe, resumiendo los test realizados en cada uno de los sprint, realizando gráficas explicativas, agregando los defectos.





Proyecto Integrador

No Environment, Today, 10:40 am

Run Again

+ New Run

Export Resu

All Tests

Passed (13)

Failed (0)

Skipped (0)

View Summ

Iteration 1

GET

Get-User

{{(BASE\_URL)}}/users

/ User / Get-User

200 OK

208 ms

2.168 KB

Pass

Status code is 200

Pass

Verificar que el primer usuario es admin

GET

Get-Booking

{{(BASE\_URL)}}/bookings

/ Booking / Get-Booking

200 OK

117 ms

3.404 KB

Pass

Status code is 200

Pass

Verificar creacion de reserva

Pass

Response time is less than 500ms

GET

Get-Booking-idProduct

{{(BASE\_URL)}}/bookings/product/15

/ Booking / Get-Booking-idProduct

200 OK

103 ms

766 B

Pass

Status code is 200

POST

Post-Booking

{{(BASE\_URL)}}/bookings

/ Booking / Post-Booking

201 Created

222 ms

549 B

Pass

Status code is 201

GET

Get-Products-Dates

{{(BASE\_URL)}}/products/booking/2022-09-12/2022-09-16

/ Booking / Get-Products-Dates

200 OK

239 ms

35.75 KB

Pass

Status code is 200

GET

Get-Products-Date-city

{{(BASE\_URL)}}/products/booking/2022-09-12/2022-09-16/3

/ Booking / Get-Products-Date-city

200 OK

154 ms

7.493 KB

Pass

Status code is 200

POST

Post-Login

{{(BASE\_URL)}}/auth/signin

/ Auth / Post-Login

200 OK

284 ms

960 B

Pass

Status code is 200

POST

Post-Register

{{(BASE\_URL)}}/auth/signup

/ Auth / Post-Register

201 Created

263 ms

471 B

Pass

Status code is 201

Pass

Verificar mensaje de body

POST

Login-negative-test

{{(BASE\_URL)}}/auth/signin

/ Auth / Login-negative-test

401 Unauthorized

109 ms

404 B

Pass

Status code is 401

## Informe de defectos

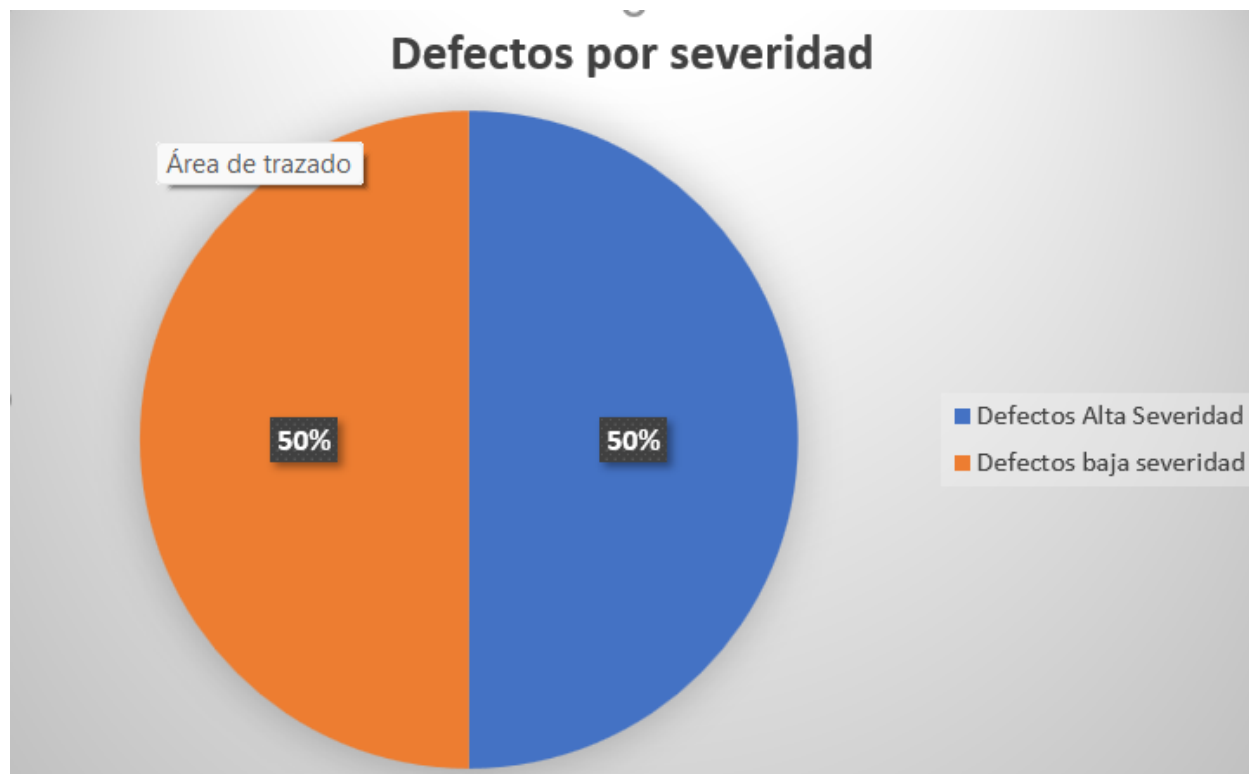
Se encontraron 4 defectos en el desarrollo de los casos de prueba, testing exploratorio, manual automatizado; los cuales se clasificaron por prioridad y severidad. Estos a su vez fueron corregidos en el desarrollo del proyecto y están finalizados sin novedad negativa.



## Prioridad



## Severidad



## Exit Criteria

Para terminar las pruebas se plantean los siguientes criterios:

- No tener abiertos defectos de severidad y/o bloqueantes.
- La ejecución de pruebas automatizadas debe tener 90% de test aprobados.
- La ejecución de test en postman debe tener el 90% aprobados.
- La ejecución de test manuales y de regresión deben tener el 90% de aprobación.
- La ejecución de pruebas exploratorias debe tener el 90% de aprobación.



## Pruebas Ejecutadas

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Prueba estática	si	si	si	si
Prueba Exploratoria	si	si	si	si
Prueba de Humo	si	si	si	si
Prueba de Regresión	si	si	si	si
Prueba postman	si	si	si	si

## Diagnóstico Final

En el desarrollo del proyecto integrador hemos realizado múltiples tareas, las cuales han sido secuenciales; estas buscaron la construcción de una API y una página web que la consumiera. El proyecto se ha iniciado desde cero, al principio se desarrollaron las funcionalidades enfocadas a la estructura general del proyecto y los estilos para brindar el good will, y luego se anexaron desde el backend la configuración requerida para almacenar los datos de los productos, y relacionar estos con las reservas del usuario o administrador, gracias a una capa de seguridad, entre otras funcionalidades. Todo ello levantado en una infraestructura de AWS.

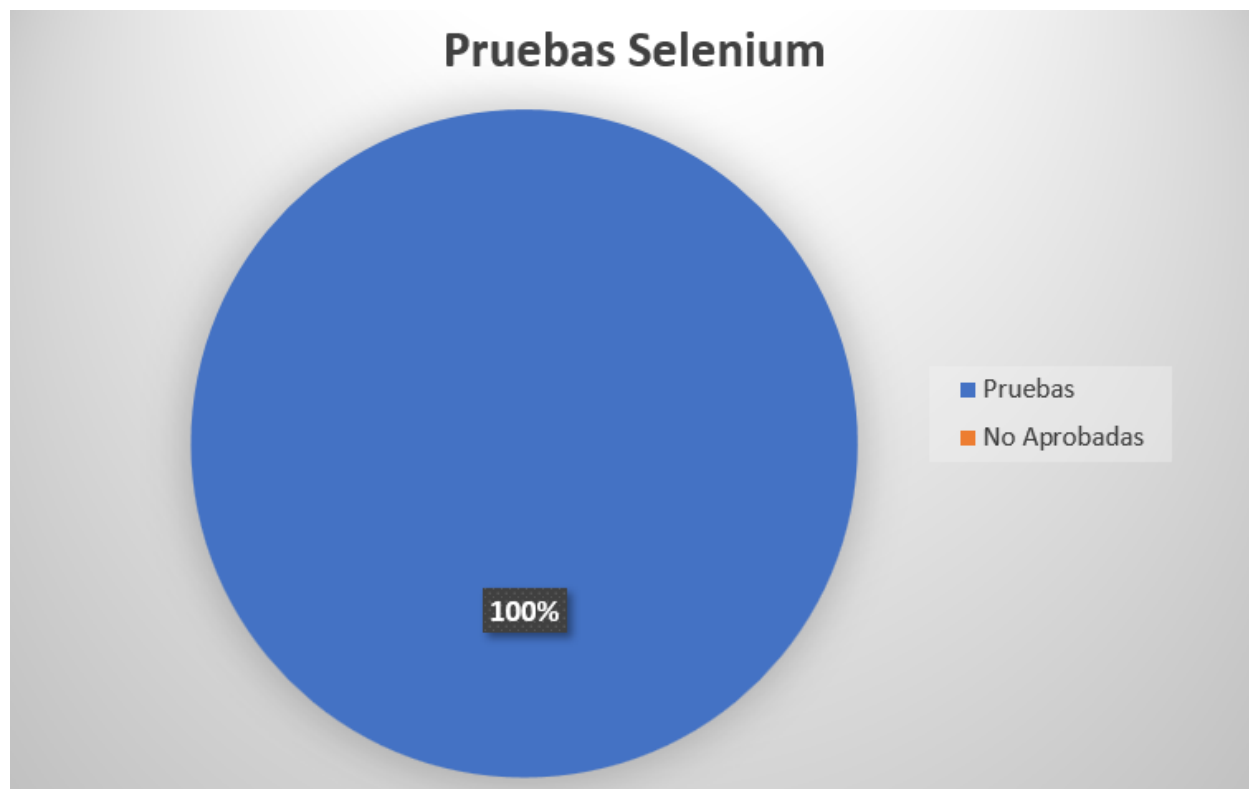
## Pruebas exploratorias

Se alcanza el 100% de efectividad en las pruebas realizadas luego de resolver errores presentados.



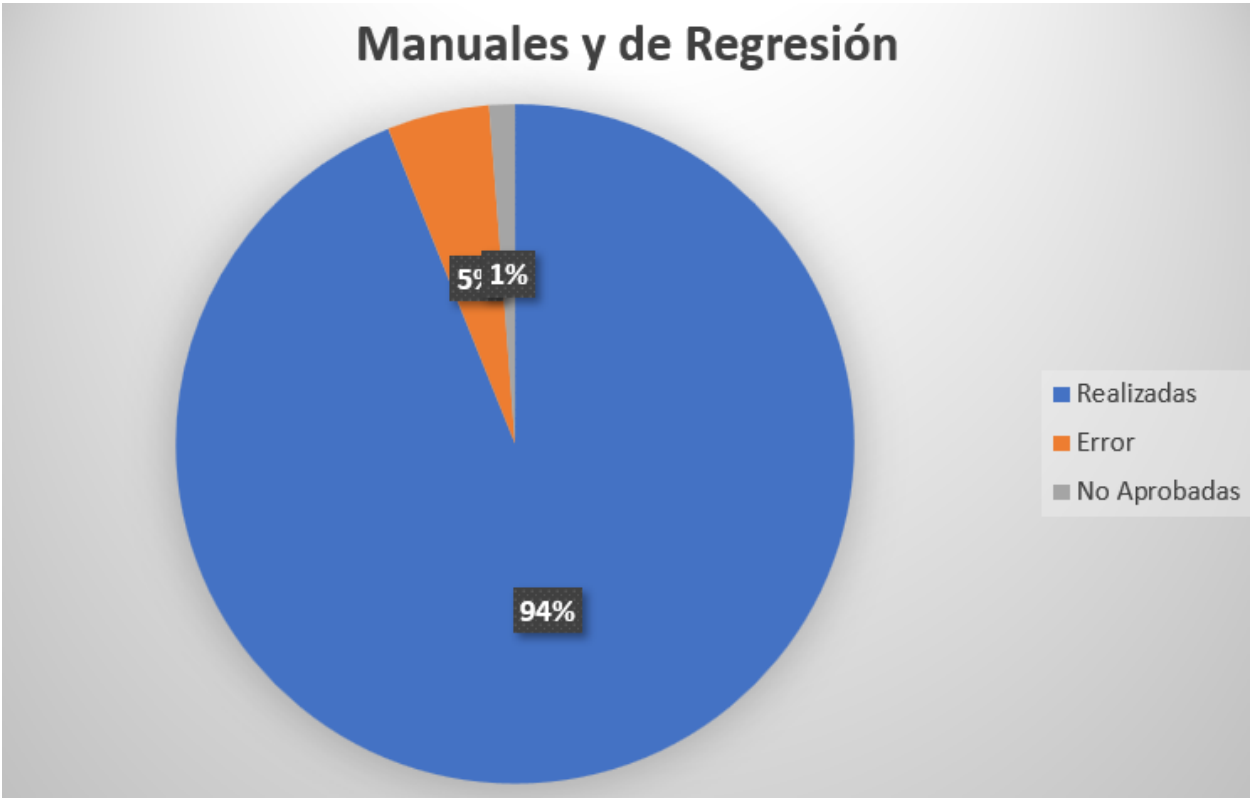
## Pruebas con Selenium

Se alcanza el 100% de efectividad en las pruebas realizadas luego de resolver errores presentados.



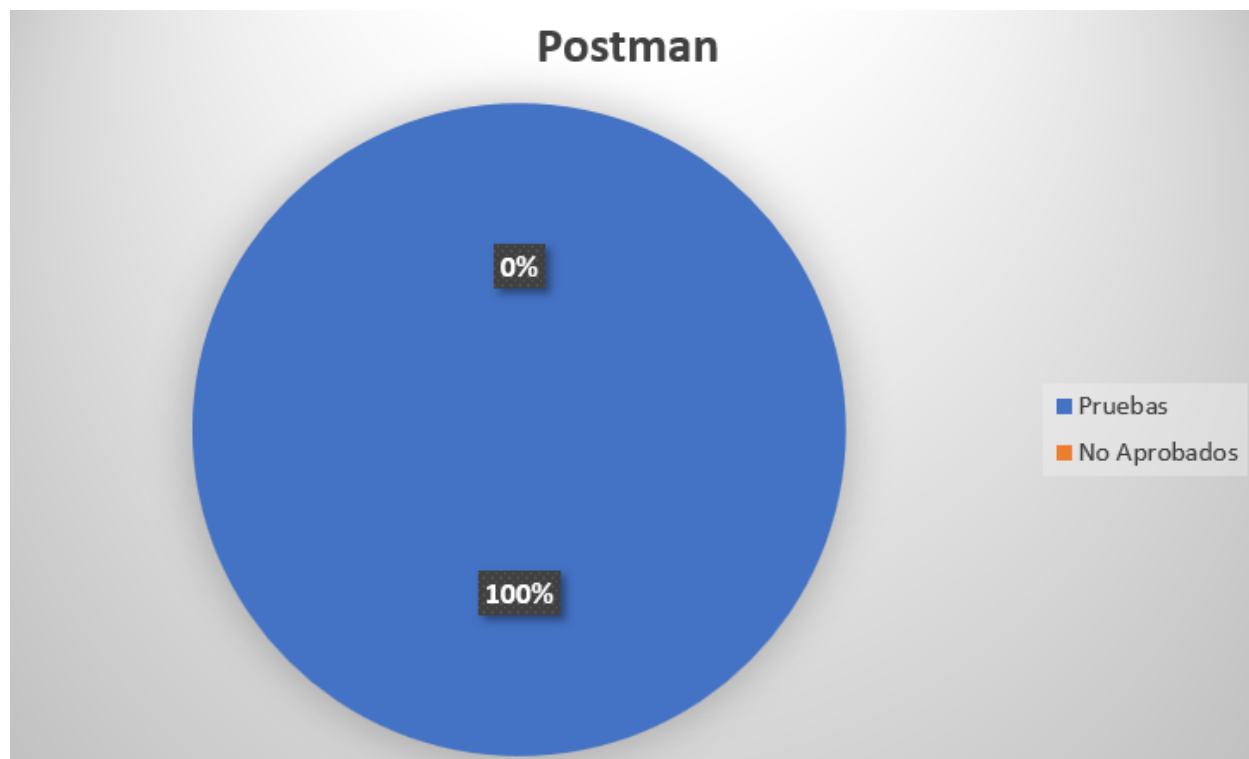
## Pruebas Manuales y de regresión

Aquí tenemos el 1% de pruebas no aprobadas después de corregir los errores presentados, dando una efectividad del 99% al final.



## Postman

Se alcanza el 100% de efectividad en las pruebas realizadas luego de resolver errores presentados.



## Porcentaje Finalizado

En este proceso se crearon 115 casos de prueba a lo largo de los 4 sprint, teniendo la ejecución de 100 de los casos de pruebas planteados, con porcentaje de cobertura del 88%, sin embargo la aceptación de las pruebas por sección género la cobertura del 90% en adelante.





Informe Casos elaborados

