

## PUNTO 1

- Plan de Pruebas

- ✓ Alcance

El alcance que tendremos dentro de las pruebas a realizar sobre la aplicación web <https://www.advantageonlineshopping.com/> se enfocará en el core del negocio en donde se realizan pruebas seleccionando productos y adicionándolos al carrito de compras, (específicamente se tendrán en cuenta los productos de la pestaña popular ítems) sobre los cuales realizaremos el proceso de compra y validaremos que lo que se muestre en el carrito de compra corresponda correctamente a los ítems adicionados, teniendo en cuenta las características de los mismos (Color y Cantidad).

Se implementarán pruebas automáticas end to end en donde se validará el proceso core del negocio, adicionando un item popular al carrito de compras, cambiándole el color y validando que lo que muestre el carrito de compras corresponda a lo ingresado.

Adicionalmente se revisarán los tiempos de respuesta de la aplicación al momento de realizar los siguientes procesos:

Login

Registro

Selección de Producto

Adición de un Producto al Carrito

Modificar características de un Producto

Comprar lo ingresado en el carrito

Escenarios o funcionalidades que no se encuentren relacionadas dentro del alcance no se tendrán en cuenta para su verificación en el proceso de pruebas que se está implementando.

- ✓ Objetivos

- Implementar pruebas funcionales de software en donde aseguremos la calidad del producto en las funcionalidades definidas
    - Desarrollar pruebas de software automáticas para asegurar la calidad en el funcionamiento core de la aplicación.
    - Garantizar el funcionamiento de la aplicación el cual sea acorde a lo esperado por el usuario

- ✓ Tipos de Pruebas

Teniendo en cuenta los Cuadrantes de pruebas, en esta prueba técnica emplearemos las siguientes pruebas:

- Q1 – No se implementarán pruebas en este cuadrante

- Q2 – Se implementarán pruebas E2E utilizando Selenium como herramienta de automatización
  - Q3 – Se implementarán pruebas Funcionales Manuales con el objetivo de revisar y validar la calidad del producto
  - Q4 – Se implementarán pruebas de Performance validando tiempos de respuesta de las funcionalidades criticas (Es posible realizar pruebas de carga y Stress pero estas no serán tenidas en cuenta dentro del alcance ya que se desconoce los requerimientos no funcionales y necesidades del producto)
- Escenarios y Casos de Prueba
  - Reporte de Bugs
  - Automatización Casos
  - Ejecución Casos

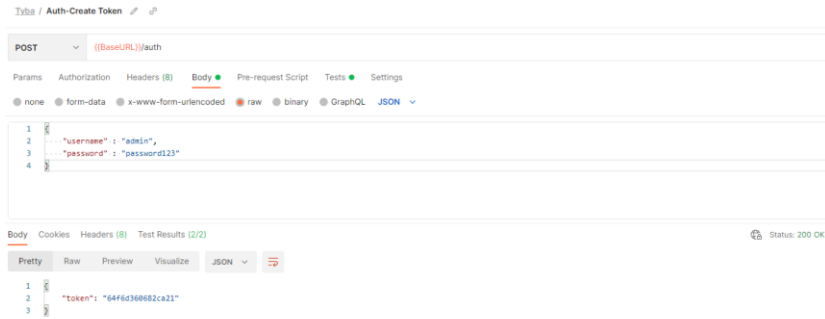
## PUNTO 2

- Setup (Creación de variables de ambiente utilizadas durante los test, las cuales permiten que las pruebas se puedan ejecutar siempre que se deseen sin tener que modificar datos en los test manualmente) adicionalmente permiten facilidad de ejecución de pruebas en diferentes ambientes

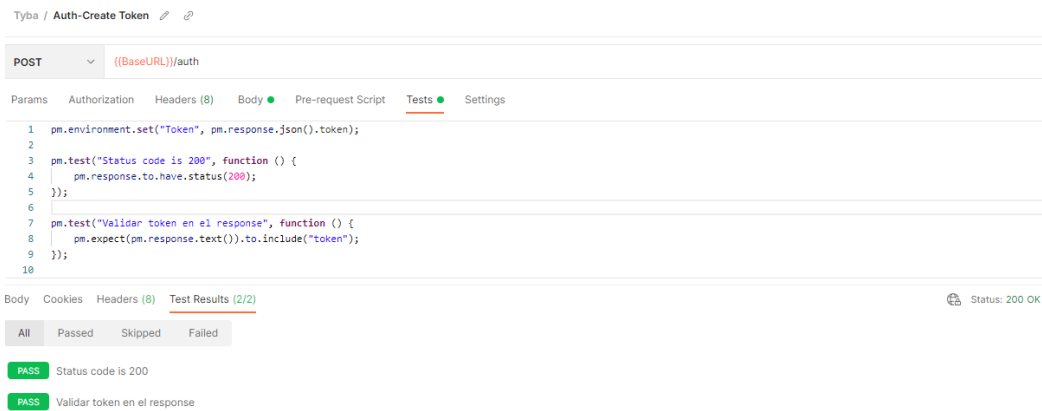
Booking

|                                     | VARIABLE           | TYPE ①  | INITIAL VALUE ①   | CURRENT VALUE ①   |
|-------------------------------------|--------------------|---------|---|---|
| <input checked="" type="checkbox"/> | BaseURL            | default | ▼ <a href="https://restful-booker.herokuapp.com">https://restful-booker.herokuapp.com</a> | <a href="https://restful-booker.herokuapp.com">https://restful-booker.herokuapp.com</a> |
| <input checked="" type="checkbox"/> | Token              | default | ▼   | 64f6d360682ca21   |
| <input checked="" type="checkbox"/> | firstname          | default | ▼ Andres  | Andres  |
| <input checked="" type="checkbox"/> | lastname           | default | ▼ Gonzalez  | Gonzalez  |
| <input checked="" type="checkbox"/> | totalprice         | default | ▼ 500   | 500   |
| <input checked="" type="checkbox"/> | depositpaid        | default | ▼ true  | true  |
| <input checked="" type="checkbox"/> | checkin            | default | ▼ 2018-01-01  | 2018-01-01  |
| <input checked="" type="checkbox"/> | checkout           | default | ▼ 2018-01-30  | 2018-01-30  |
| <input checked="" type="checkbox"/> | additionalneeds    | default | ▼ Breakfast   | Breakfast   |
| <input checked="" type="checkbox"/> | bookingid          | default | ▼   | 5815  |
|                                     | Add a new variable |         |   |   |

- Creación de token



Se utiliza una variable de ambiente para guardar el token generado y usarlo automáticamente en los otros servicios.



Adicionalmente se hacen test para validar el status code y que exista un token en el response.

- Crear Booking

Se ejecuta el test de creación del Booking con los datos previamente almacenados en las variables de ambiente

Tyba / Create Booking

POST ▼ `{{BaseURL}}/booking`

Params Authorization Headers (8) **Body** ● Pre-request Script Tests ● Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼

```

1  {
2    "firstname": "{{firstname}}",
3    "lastname": "{{lastname}}",
4    "totalprice": {{totalprice}},
5    "depositpaid": {{depositpaid}},
6    "bookingdates": {
7      "checkin": "{{checkin}}",
8      "checkout": "{{checkout}}"
9    },
10   "additionalneeds": "{{additionalneeds}}"
11 }

```

ody Cookies Headers (8) Test Results (2/2) 🔍 Status: 200 OK

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```

1  {
2    "bookingid": 5946,
3    "booking": {
4      "firstname": "Andres",
5      "lastname": "Gonzalez",
6      "totalprice": 500,
7      "depositpaid": true,
8      "bookingdates": {
9        "checkin": "2018-01-01",
10       "checkout": "2018-01-30"
11     },
12     "additionalneeds": "Breakfast"
13   }
14 }

```

En esta caso se utiliza el Bookingid para guardarlo en una variable y utilizarlo posteriormente en los servicios de consulta, actualización y eliminación de tal forma que permita ser reutilizable la collection.

Tyba / Create Booking

POST ▼ `{{BaseURL}}/booking`

Params Authorization Headers (8) **Body** ● Pre-request Script **Tests** ● Settings

```

1  pm.environment.set("bookingid", pm.response.json().bookingid);
2
3  pm.test("Status code is 200", function () {
4    pm.response.to.have.status(200);
5  });
6
7  pm.test("Validar booking creado", function () {
8    pm.expect(pm.response.json().booking.firstname).to.eql(pm.environment.get("firstname"));
9    pm.expect(pm.response.json().booking.lastname).to.eql(pm.environment.get("lastname"));
10   pm.expect(pm.response.json().booking.totalprice.toString()).to.eql(pm.environment.get("totalprice"));
11   pm.expect(pm.response.json().booking.depositpaid.toString()).to.eql(pm.environment.get("depositpaid"));
12   pm.expect(pm.response.json().booking.bookingdates.checkin).to.eql(pm.environment.get("checkin"));
13   pm.expect(pm.response.json().booking.bookingdates.checkout).to.eql(pm.environment.get("checkout"));
14   pm.expect(pm.response.json().booking.additionalneeds).to.eql(pm.environment.get("additionalneeds"));
15 });

```

Body Cookies Headers (8) Test Results (2/2) 🔍 Status: 200 OK 1

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```

1  {
2    "bookingid": 5946,
3    "booking": {
4      "firstname": "Andres",
5      "lastname": "Gonzalez",
6      "totalprice": 500,
7      "depositpaid": true,
8      "bookingdates": {
9        "checkin": "2018-01-01",
10       "checkout": "2018-01-30"
11     },
12     "additionalneeds": "Breakfast"
13   }
14 }

```

En la creación se valida el response contra los datos ingresados, adicionalmente del status code de la respuesta del servicio.

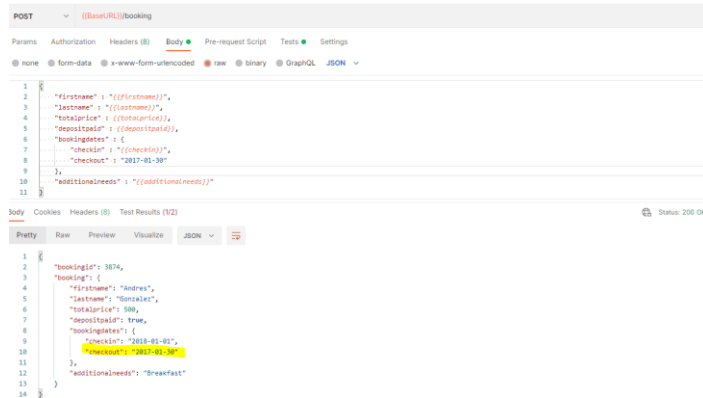
Se encuentran los siguientes Issues:

- ✓ En el servicio de creación no se esta solicitando el token de seguridad, este tipo de servicios debe tener seguridad.

<https://restful-booker.herokuapp.com/booking>

Ver Reporte ISSUE001

- ✓ En el servicio de creación no se tiene validación de fechas, permite entonces tener una fecha de chekout inferior a la fecha del checkin. Hay escenarios que requieren tener una doble validación, back y front.



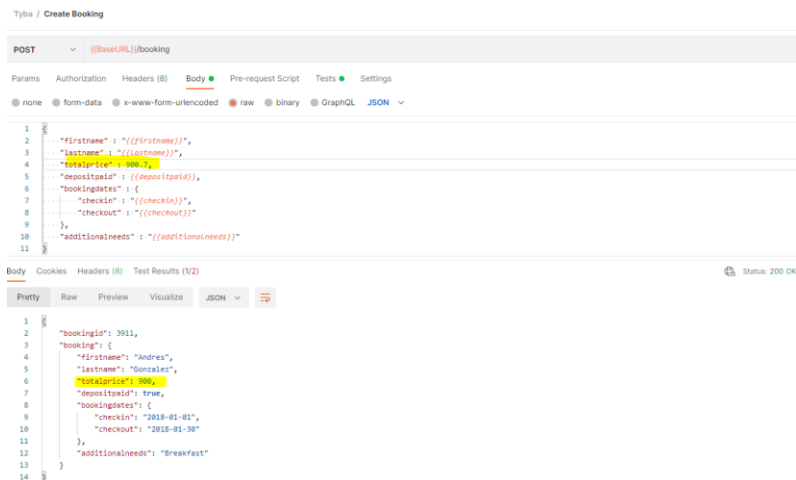
```
POST /{{BaseURL}}/booking

{
  "firstname": "{{firstname}}",
  "lastname": "{{lastname}}",
  "totalprice": {{totalprice}},
  "depositpaid": {{depositpaid}},
  "bookingdates": {
    "checkin": "{{checkin}}",
    "checkout": "2017-01-30"
  },
  "additionalneeds": "{{additionalneeds}}"
}

{
  "bookingid": 3674,
  "booking": {
    "firstname": "Andres",
    "lastname": "Gonzalez",
    "totalprice": 590,
    "depositpaid": true,
    "bookingdates": {
      "checkin": "2018-01-01",
      "checkout": "2017-01-30"
    },
    "additionalneeds": "Breakfast"
  }
}
```

Ver Reporte ISSUE002

- ✓ En el servicio de creación cuando se envia un totalprice el cual contiene decimales, no respeta este valor sino que almacena en el booking un valor entero. Se debe revisar el requerimiento para validar la necesidad



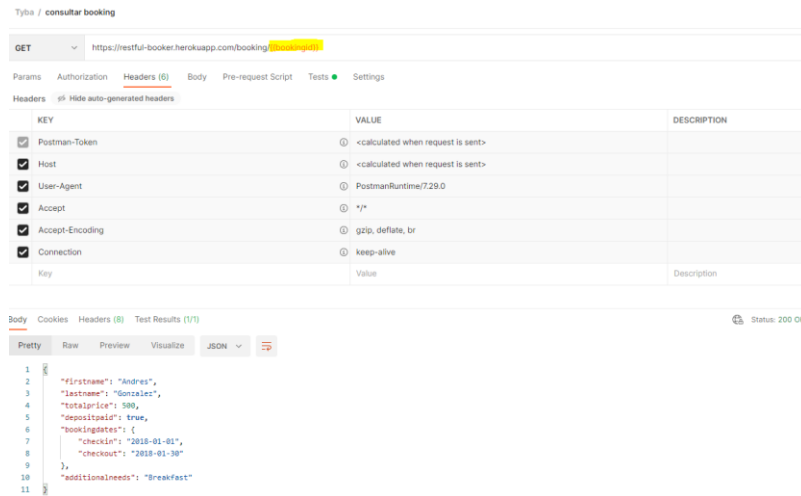
```
POST /{{BaseURL}}/booking

{
  "firstname": "{{firstname}}",
  "lastname": "{{lastname}}",
  "totalprice": 590.2,
  "depositpaid": {{depositpaid}},
  "bookingdates": {
    "checkin": "{{checkin}}",
    "checkout": "{{checkout}}"
  },
  "additionalneeds": "{{additionalneeds}}"
}

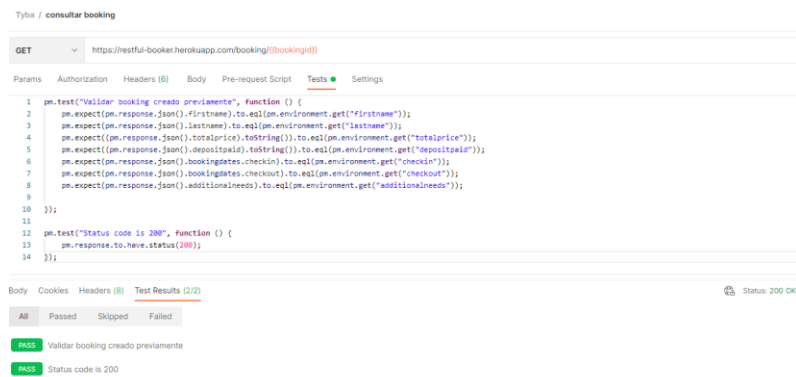
{
  "bookingid": 3911,
  "booking": {
    "firstname": "Andres",
    "lastname": "Gonzalez",
    "totalprice": 590,
    "depositpaid": true,
    "bookingdates": {
      "checkin": "2018-01-01",
      "checkout": "2018-01-30"
    },
    "additionalneeds": "Breakfast"
  }
}
```

- Consultar Booking

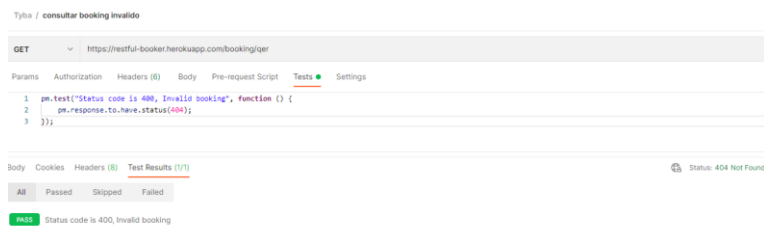
Se realiza consumo del servicio de consulta de booking en la cual utilizamos el Bookingid devuelto en el servicio de creación.



Se valida que la data que responda el servicio sea igual a los datos con los cuales se creó el booking, adicionalmente de validar el status code del response.



Se valida que el servicio solo retorne valores cuando se consulta un booking existente



Se encuentran los siguientes Issues:

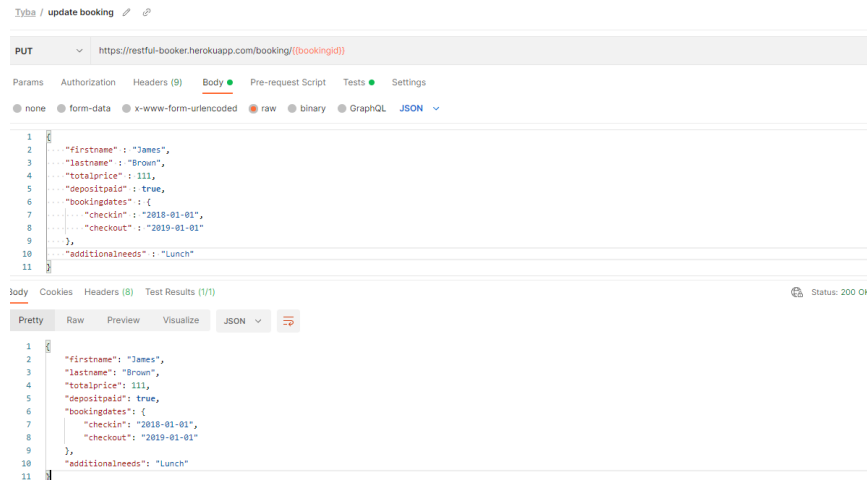
- ✓ En el servicio de consulta no se está solicitando el token de seguridad, este tipo de servicios debe tener seguridad.

Se anexa servicio al reporte hecho previamente.

Ver Reporte ISSUE001

- Actualizar Booking

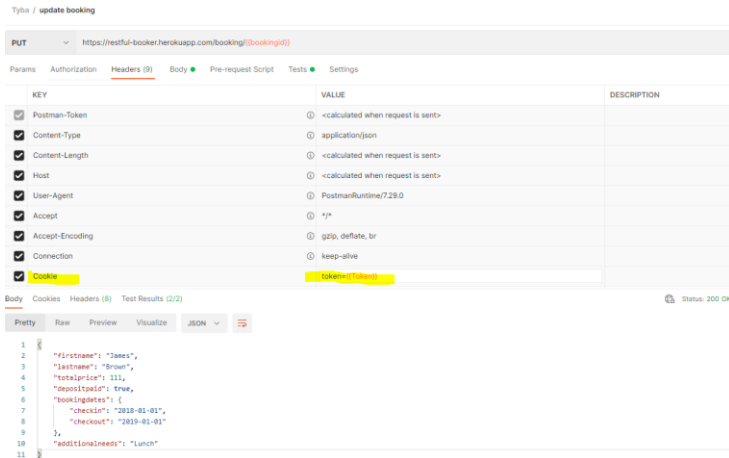
Se realiza consumo del servicio de actualización de booking en la cual utilizamos el Bookingid devuelto en el servicio de creación y modificamos todos los datos para validar que efectivamente se actualice todo correctamente.



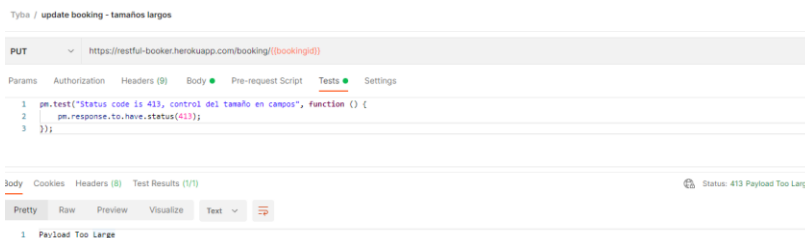
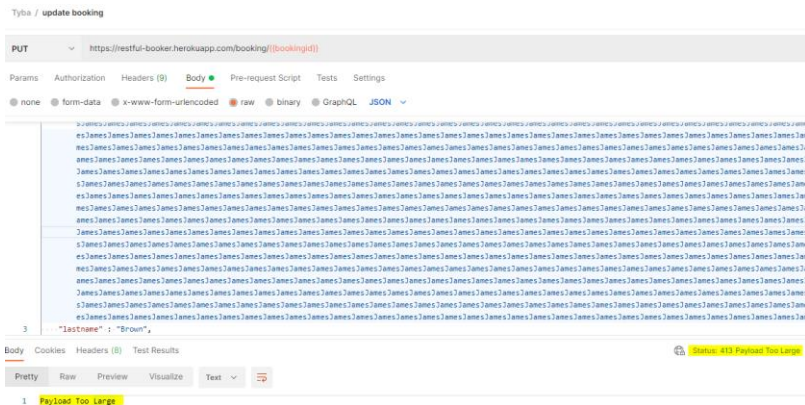
Se crean test que validen que la información del response corresponda a la información modificada.



Se valida correctamente que valide el token de seguridad creado previamente



Se realizan validaciones del tamaño de los campos y el control sobre esto, es importante para no tener un costo muy alto en base de datos o validar que los servicios no respondan información relevante en la estructura (Se debe validar los valores limites de acuerdo al requerimiento o historia de usuario y validar que efectivamente cumplan con los tamaños permitidos)



- Eliminar Booking

Se realiza el consumo del servicio de eliminación de booking creado previamente, para esto se utiliza el bookingId devuelto en el servicio de creación.



Tyba / delete booking

DELETE ▼ https://restful-booker.herokuapp.com/booking/bookingid

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Headers ⌵ Hide auto-generated headers

| KEY   | VALUE                               | DESCRIPTION |
|---|-------------------------------------|-------------|
| <input checked="" type="checkbox"/> Postman-Token   | ① <calculated when request is sent> |             |
| <input checked="" type="checkbox"/> Host            | ① <calculated when request is sent> |             |
| <input checked="" type="checkbox"/> User-Agent      | ① PostmanRuntime/7.29.0             |             |
| <input checked="" type="checkbox"/> Accept          | ① *                                 |             |
| <input checked="" type="checkbox"/> Accept-Encoding | ① gzip, deflate, br                 |             |
| <input checked="" type="checkbox"/> Connection      | ① keep-alive                        |             |
| <input checked="" type="checkbox"/> Cookie          | token=({Token})                     |             |
| Key   | Value                               | Description |

Body Cookies Headers (8) Test Results ⌵ Status: 201 Created

Pretty Raw Preview Visualize Text ⌵

1 Created

Se realiza prueba en la cual se intenta eliminar nuevamente el mismo booking que ya se había eliminado.

Tyba / delete booking No existente

DELETE ▼ https://restful-booker.herokuapp.com/booking/({bookingid})

Params Authorization Headers (7) Body Pre-request Script Tests ● Settings

```

1 pm.test("Status code is 405 - Eliminar booking que no exista", function () {
2   pm.response.to.have.status(405);
3 });

```

Body Cookies Headers (8) Test Results (1/1) ⌵ Status: 405 Method Not Allowed

Pretty Raw Preview Visualize Text ⌵

1 Method Not Allowed

Se valida que el servicio requiera el token para poder realizar la eliminación del booking

Tyba / delete booking sin token

DELETE ▼ https://restful-booker.herokuapp.com/booking/({bookingid})

Params Authorization Headers (7) Body Pre-request Script Tests ● Settings

```

1 pm.test("Status code is 403 - Eliminar booking sin usar token", function () {
2   pm.response.to.have.status(403);
3 });

```

Body Cookies Headers (8) Test Results (1/1) ⌵ Status: 403 Forbidden

Pretty Raw Preview Visualize Text ⌵

1 Forbidden

## EJECUCION DE LOS TEST

- ✓ Se realiza ejecución de toda la colección y se valida que corran todos los test correctamente.

Tyba Booking, Today, 7:41 am

| RUN SUMMARY |                                 | 1     |
|-------------|---------------------------------|-------|
| ▶ POST      | Auth-Create Token               | 2   0 |
| ▶ POST      | Create Booking                  | 2   0 |
| ▶ GET       | consultar booking               | 2   0 |
| ▶ GET       | consultar booking invalido      | 1   0 |
| ▶ PUT       | update booking                  | 2   0 |
| ▶ PUT       | update booking - tamaños largos | 1   0 |
| ▶ DELETE    | delete booking                  | 1   0 |
| ▶ DELETE    | delete booking No existente     | 1   0 |
| ▶ DELETE    | delete booking sin token        | 1   0 |

- ✓ Adicionalmente se valida la ejecución que corra en varias iteraciones para garantizar que la prueba si sea mantenible y reutilizable con todos los datos cargados en el setup

### Run Settings

Iterations

Delay  ms

Data

Advanced settings

- ☐ Save responses ⓘ
- ☒ Keep variable values ⓘ
- ☐ Run collection without using stored cookies
- ☒ Save cookies after collection run ⓘ

RUN SUMMARY

|  |    | 1 | 2 | 3 | 4 | 5 |
|--|----|---|---|---|---|---|
| ▶ <b>POST</b> Auth-Create Token              | 10 | 0 |   |   |   |   |
| ▶ <b>POST</b> Create Booking                 | 11 | 0 |   |   |   |   |
| ▶ <b>GET</b> consultar booking               | 10 | 0 |   |   |   |   |
| ▶ <b>GET</b> consultar booking invalido      | 5  | 0 |   |   |   |   |
| ▶ <b>PUT</b> update booking                  | 10 | 0 |   |   |   |   |
| ▶ <b>PUT</b> update booking - tamaños largos | 5  | 0 |   |   |   |   |
| ▶ <b>DELETE</b> delete booking               | 5  | 0 |   |   |   |   |
| ▶ <b>DELETE</b> delete booking No existente  | 5  | 0 |   |   |   |   |
| ▶ <b>DELETE</b> delete booking sin token     | 5  | 0 |   |   |   |   |