

# TEST CASE DEVELOPMENT.

## Casillas de los Test Case:

1. **ID:** Used as a reference number or label to efficiently locate and manage test cases
2. **Descriptions:** A brief, clear summary of what will be tested
3. **Test date:** Data that will be used in the test
4. **Steps:** the step-by-step process to recreate the test case accurately
5. **Expected Results:** Explanation of how the test should perform
6. **Obtained Results:** Explanation of how the test performed

## Preconditions Frontend:

1. Have internet access
2. Run the local project
3. Access the browser
4. Navigate to: <http://localhost:3000>

## *Auth and Dashboard*

ID	Descriptions	Test Date	Steps	Expected result	Obtained result
AD-01	Confirm and ensure that the login is successful using correct credentials	<b>1. username:</b> testuser <b>2. password:</b> password	<b>1.</b> Click on the Username container. <b>2.</b> Enter the username indicated in Test Data. <b>3.</b> Click on the Password container. <b>4.</b> Enter the password indicated in Test Data <b>5.</b> Click on the Login.	The Login is successful and the message "Logged in with token: sampletoken" is displayed on the screen.	The login was successful, and the expected message appears

AD-02	Confirm and ensure that the login fails when invalid credentials are used, especially when the username is empty	<b>1.username:</b> <b>2. password:</b> password	<b>1.</b> Click on the Username container. <b>2.</b> Enter the username indicated in Test Data. <b>3.</b> Click on the Password container. <b>4.</b> Enter the password indicated in Test Data <b>5.</b> Click on the Login.	The Login is not successful and the message "Login failed" is displayed on the screen	The login was not successful; the message 'login failed' is displayed
AD-03	Confirm that the login does not proceed with invalid credentials, particularly when the password is empty	<b>1. username:</b> testuser <b>2. password:</b>	<b>1.</b> Click on the Username container. <b>2.</b> Enter the username indicated in Test Data. <b>3.</b> Click on the Password container. <b>4.</b> Enter the password indicated in Test Data <b>5.</b> Click on the Login.	The Login is not successful and the message "Login failed" is displayed on the screen	The login was not successful; the message 'login failed' is displayed
AD-04	Ensure that the login is not successful when both credential parameters are empty.	<b>1.username:</b> <b>2. password:</b>	<b>1.</b> Click on the Username container. <b>2.</b> Enter the username indicated in Test Data. <b>3.</b> Click on the Password container. <b>4.</b> Enter the password indicated in Test Data <b>5.</b> Click on the Login.	The Login is not successful and the message "Login failed" is displayed on the screen	The login was not successful; the message 'login failed' is displayed
AD-05	Confirm that the login is not successful when an invalid username is used	<b>1. username:</b> testuser2324 <b>2. password:</b> password	<b>1.</b> Click on the Username container. <b>2.</b> Enter the username indicated in Test Data. <b>3.</b> Click on the Password container. <b>4.</b> Enter the password indicated in Test Data <b>5.</b> Click on the Login.	The Login is not successful and the message "Login failed" is displayed on the screen	The login was not successful; the message 'login failed' is displayed
AD-06	Ensure that the login fails when an invalid password is entered	<b>1. username:</b> testuser <b>2. password:</b> password1234	<b>1.</b> Click on the Username container. <b>2.</b> Enter the username indicated in Test Data. <b>3.</b> Click on the Password container. <b>4.</b> Enter the password indicated in Test Data <b>5.</b> Click on the Login.	The Login is not successful and the message "Login failed" is displayed on the screen	The login was not successful; the message 'login failed' is displayed
AD-07	Confirm that the logout is successfully completed when the user is logged in	<b>1. username:</b> testuser <b>2. password:</b> password	1- Click on the Logout link.	The Logout is successful and is redirected to the initial page.	There is no option available to log out on the website

AD-08	Ensure that access to the Dashboard page is successful when the user is authenticated.	<b>1. username:</b> testuser <b>2. password:</b> password	1- Click on the Dashboard link.	The link works and is redirected to the dashboard page and can see the title "Dashboard" and the message "Welcome to the user dashboard!"	Access to the Dashboard page was successful, and the expected message appears successfully.
AD-09	Validate and verify that access to the Dashboard page is successful without being logged in.	<b>Not data</b>	1- Click on the Dashboard link.	The link works and is redirected to the dashboard page and can see the title "Dashboard" and the message "Welcome to the user dashboard!"	Access to the Dashboard page was successful, and the expected message appears successfully.

## **Product and Order**

ID	Descriptions	Test Date	Steps	Expected result	Obtained result
PO-01	Confirm that access to the Products page is successful when the user is properly authenticated	<b>1. username:</b> testuser <b>2. password:</b> password	<b>1- Click on the Products link.</b>	The link works and is redirected to the Products page and can see the title "Product List" and below the list of Products created.	Access to the Products section has been successful.
PO-02	Validate and verify that there are no products loaded, when the application is just executed.	Not data	<b>1- Click on the Products link.</b>	The link works and is redirected to the Products page and can see the title "Product List" and below the empty list of Products.	The empty list is displayed successfully.
PO-03	Validate and verify that the products are observed successfully	<b>1.</b> Products must be created. <b>2.</b> API access must be available.	<b>1- Click on the Products link.</b>	The link works and is redirected to the Products page and can see the title "Product List" and below the list of Products with some product	The product list is displayed successfully.

<b>PO-04</b>	Ensure that when a new product is added through the API, it appears on the screen after refreshing	ID: 1 name: giftcard price: 20	<ol style="list-style-type: none"> <li>1. Click on the Products link, and looking the products.</li> <li>2. Create a new prduct from the API, with the indicated fields of Test data.</li> <li>3. Go back to the browser and refresh the Products page.</li> </ol>	Verify if the new product created is there.	The product has been successfully created.
<b>PO-05</b>	Confirm that when a product is deleted through the API, the deletion is reflected on the screen after refreshing.	DELETE: ID: 1	<ol style="list-style-type: none"> <li>1. Click on the Products link and review the listed products.</li> <li>2. Delete the specified product using the API with the provided test data.</li> <li>3. Return to the browser and refresh the Products page to confirm that the changes are reflected.</li> </ol>	Verify that the product no longer exists in the list.	The product has been successfully removed.
<b>PO-06</b>	Ensure that when the values of a product are modified through the API, the changes are correctly displayed on the screen after refreshing.	Update the price of the product with Id: 0 from 10 to 20 through the API	<ol style="list-style-type: none"> <li>1. Click on the Products link and review the list of products.</li> <li>2. Update the product using the API with the specified values from the test data.</li> <li>3. Return to the browser and refresh the Products page to verify that the changes are reflected.</li> </ol>	Verify if the changes in the product are visible.	The product has been successfully updated.
<b>PO-07</b>	Confirm that access to the Orders page is successful when the user has logged in correctly	username: testuser password: password	1- Click on the Orders link.	The link works and is redirected to the Orders page and can see the title "Order List" and below the list of Orders created.	Access to the Orders section has been successful.
<b>PO-08</b>	Ensure that the order is displayed correctly on the screen, with all details visible and well-organized	<ol style="list-style-type: none"> <li>1. Orders must be created.</li> <li>2. API access must be available.</li> </ol>	1- Click on the Orders link.	The link works and is redirected to the Orders page and can see the title "Orders List" and below the list of Order with some order.	The order list is displayed successfully.

<b>PO-09</b>	Confirm that when a new order is added through the API, it appears on the screen after refreshing	id: 1 productName: giftcard pro quantity: 1 status: Delivered	<ol style="list-style-type: none"> <li>1. Click on the Orders link, and looking the orders.</li> <li>2. Create a new order from the API, with the indicated fields of Test data.</li> <li>3. Go back to the browser and refresh the Orders page.</li> </ol>	Verify if the new order created is there.	The order has been successfully created.
<b>PO-10</b>	Asegúrate de que, al eliminar un pedido mediante la API, la eliminación se muestre en la pantalla después de actualizarla.	DELETE: ID: 1	<ol style="list-style-type: none"> <li>1- Click on the Orders link, and looking the orders.</li> <li>2- Delete the order indicated on Test data from the API.</li> <li>3- Go back to the browser and refresh the Orders page.</li> </ol>	Verify that the order no longer exists in the list.	The order has been successfully removed.
<b>PO-11</b>	Confirm that when the values of an order are modified through the API, the changes are correctly reflected on the screen after refreshing	Change the status of the order with the specified Id from 'In progress' to 'Delivered' using the API	<ol style="list-style-type: none"> <li>1. Click on the Orders link, and looking the orders.</li> <li>2. Update the order from the API, with the indicated fields of Test data.</li> <li>3. Go back to the browser and refresh the Orders page.</li> </ol>	Verify if the changes in the order are visible.	The order hasn't been updated in the list.

### Preconditions for Backend testing:

1. Have internet access
2. Access the browser
3. Open Postman
4. Run the local project

## *All backend*

ID	Descriptions	Test date	steps	Expected result	Obtained result
AB-01	Verify that users can successfully log in with valid credentials.	<b>1. username:</b> testuser <b>2.password:</b> password	<b>1-</b> Enter the username and password indicated in Test Data. <b>2-</b> Click on the Send button.	The response has the status code 200 OK and the body is { "token": "sampletoken" }	The login was successful and the token is displayed in the body. The status code and response time are correct.
AB-02	Validate that the system rejects login with incorrect credentials.	<b>1. username:</b> testuser12 <b>2.password:</b> password	<b>1-</b> Enter the username and password indicated in Test Data. <b>2-</b> Click on the Send button.	The response has the status code 401 Unauthorized and the body shows error information.	The login was not executed and a corresponding error message is displayed in the body. The status code and response time are correct.
AB-03	Validate and verify that the login isn't successful with invalid credentials. Username empty.	<b>1. username:</b>  <b>2.password:</b> password	<b>1-</b> Enter the username and password indicated in Test Data. <b>2-</b> Click on the Send button.	The response has the status code 401 Unauthorized and the body shows error information.	The login was not executed and a corresponding error message is displayed in the body. The status code and response time are correct.
AB-04	Validate and verify that the login isn't successful with invalid credentials. Password empty.	<b>1. username:</b> testuser <b>2.password:</b>  	<b>1-</b> Enter the username and password indicated in Test Data. <b>2-</b> Click on the Send button.	The response has the status code 401 Unauthorized and the body shows error information.	The login was not executed and a corresponding error message is displayed in the body. The status code and response time are correct.
AB-05	Validate and verify that the login isn't successful with invalid credentials. Without the username parameter.	<b>2.password:</b> password	<b>1-</b> Enter the username and password indicated in Test Data. <b>2-</b> Click on the Send button.	The response has the status code 401 Unauthorized and the body shows error information.	The login was not executed and a corresponding error message is displayed in the body. The status code and response time are correct.
AB-06	Validate and verify that the login isn't successful with invalid credentials. Without the Password parameter.	<b>1. username:</b> testuser	<b>1-</b> Enter the username and password indicated in Test Data. <b>2-</b> Click on the Send button.	The response has the status code 401 Unauthorized and the body shows error information.	The login was not executed and a corresponding error message is displayed in the body. The status code and response time are correct.

<b>AB-07</b>	Verify that the API allows to create a product with valid data	id: 0 Name: giftcard Price: 10	<b>1-</b> Enter the indicated in Test Data. <b>2-</b> Click on the Send button.	The API responds with a success message and a 201 status code.	The API responds with a success message and a 201 status code.
<b>AB-08</b>	Verify that the API allows consulting the list of products correctly		1. Send a GET request to /api/Product.	The API responds with the complete list of products and a 200 status code.	The API responds with the complete list of products and a 200 status code.
<b>AB-09</b>	Ensure that the API does not allow the creation of a product if the provided ID already exists in the system	id: 0 Name: giftcard Price: 300	1. Send a POST request to /api/Product using an ID that already exists.	The API responds with an error message and a 409 status code, indicating that the ID already exists	If it allows creating the same product with the same ID
<b>AB-10</b>	Verify that the API allows editing a specific product using its ID.	id: 0 Name: giftcard Price: 20	1. Send a PUT request to /api/Product with a specific ID and new data to update the product	The API responds with a success message and a status code 200.	The API responds with a success message and a status code 204.
<b>AB-11</b>	Verify that the API allows you to delete a specific product using its ID.	<b>DELETE:</b> ID:0	1. Send a DELETE request /api/Product with a specific ID.	The API responds with a success message and a status code 200.	The API responds with a success message and a status code 200.
<b>AB-12</b>	Verify the behavior of the system when an attempt is made to delete a product that does not exist.		1. Send a PUT request to /api/Product with a product ID that does not exist	The API responds with an error message and 404 status code.	The API responds with an error message and 404 status code.
<b>AB-13</b>	Verify that the API correctly returns all existing orders.		1. Send a GET request to /api/Order without additional parameters	The API responds with a complete list of commands and a status code 200.	The API responds with a complete list of commands and a status code 200.
<b>AB-14</b>	Ensure that the API correctly creates a specific order when provided with a valid ID, and that the order appears in the order list after its creation	id: 1 productName: giftcard quantity: 0 status: In process	1. Send a GET request to the /api/Order with a valid order ID as parameter.	The API responds with the details of the specific order and a 200 status code	The API responds with the details of the specific order and a 200 status code
<b>AB-15</b>	Test the behavior of the API when requesting an order with an ID that does not exist.		1. Send a GET request to /api/Order with an order ID that does not exist.	he API responds with an error message and a 404 status code.	The API responds with an error message and a 404 status code.

<b>AB-16</b>	Validate that the API properly handles the creation of an order when the productName is empty	id: 2 productName: quantity: 1 status: In process	1. Send a POST request to /api/Order with empty productName in the body	The API responds with an error message and a status code 405	The API responded with a 201 status code and allowed the creation of an order without a product name
<b>AB-17</b>	Validate that the API properly handles the creation of an order when the status is empty	id: 2 productName: Giftcard quantity: 1 status:	1. Send a POST request to /api/Order with empty status in the body	The API responds with an error message and a status code 405	The API responded with a 201 status code and allowed the creation of an order without a status
<b>AB-18</b>	Validate that the API properly handles the creation of an order when the ID is empty	productName: Giftcard quantity: 1 status: in process	1. Send a POST request to /api/Order with empty ID in the body	The API responds with an error message and a status code 405	The API responded with a 201 status code and allowed the creation of an order without an ID, which was automatically assigned the ID: 0
<b>AB-19</b>	Validate that the API properly handles the creation of an order when the Quantity is empty	id: 2 productName: Giftcard status: in process	1. Send a POST request to /api/Order with empty Quantity in the body	The API responds with an error message and a status code 405	The API responded with a 201 status code and allowed the creation of an order without an Quantity, which was automatically assigned the ID: 0
<b>AB-20</b>	Validate that the API rejects the creation of an order when quantity is negative.	id: 6 productName: Gift Card quantity: -1 status: In process	1. Send a POST request to /api/Order with negative quantity in the body	The API responds with an error message and a status code 405	The API responded with a 201 status code and allowed the creation of an order with a negative quantity
<b>AB-21</b>	Update order with valid data (without changing the ID	id: 1 productName: giftcard quantity: 1 status: delivered	1. Send a PUT request to /api/Order/{id} with a body that includes productName, quantity, and status with valid values.	The API responds with a success message and a status code 200 or 204, and the order is successfully updated in the database, without changing the id	The API responded with a 204 status code and updated the order without changing the ID.



<b>AB-22</b>	Update the ID of an order		1. Send a PUT request to /api/Order/{id} with a body that includes a different id than the current one.	The API responds with an error message and a status code 400 or 409, indicating that the id is not editable.	The ID does not change, but the API does not provide an error message
<b>AB-23</b>	Validate that the API properly handles the update of an order when the productName is other	id: 1 productName: PokeCard quantity: 1 status: in process	1. Send a PUT request to /api/Order/{id} with a body that includes a different ProductName	The API responds with a success message and a status code 200	The ProductName does not change, but the API does not provide an error message
<b>AB-24</b>	Validate that the API properly handles the update of an order when the Quantity is other	id: 1 productName: PokeCard quantity: 4 status: in process	1. Send a PUT request to /api/Order/{id} with a body that includes a different Quantity	The API responds with a success message and a status code 200	The Quantity does not change, but the API does not provide an error message
<b>AB-25</b>	Validate that the API properly handles the update of an order when the Quantity is negative	id: 1 productName: PokeCard quantity: -4 status: in process	1. Send a PUT request to /api/Order/{id} with a body that includes a negative Quantity	The API responds with a success message and a status code 400	The Quantity does not change, but the API does not provide an error message
<b>AB-26</b>	Validate that the API properly handles the update of an order when the Status is other	id: 1 productName: Gift Card quantity: 1 status: delivered	1. Send a PUT request to /api/Order/{id} with a body that includes a Different Status	The API responds with a success message and a status code 200	The status changes, but the API does not provide an error message and returns a 204 status code
<b>AB-27</b>	Verify that the API allows deleting an existing order with a valid order ID.	DELETE: ID: 1	1. Send a DELETE request to /api/Order/{id}	The API responds with a success message and a status code 200	The Order delete , but the API does not provide an error message and returns a 204 status code
<b>AB-28</b>	Delete order with not existent ID	DELETE ID:19	. Send a DELETE request to /api/Order/{id} with an id that does not exist.	The API responds with a success message and a status code 404	The API responds with a success message and a status code 404

**The evidence is in the folder named 'evidence'**

