

Orinoco – Oak Furniture – Test Plan

Author: Reinaldo Ferreira

Github Repository: <https://github.com/Reinaldodsf/OrinocoOakFurniture>

1. Initial Page (index.html)

Teste 1.1. Fail AJAX Request

Given a user that opens the index.html page

When the AJAX request fails

Then user should receive an error – “Could not connect to the server! Please try again later.”

Teste 1.2. Success AJAX Request

Given a user that opens the index.html page

When the AJAX request is successful

Then the page should display all the products received from the server.

Teste 1.3. Product ‘See More Details’ Button

Given a user in the index.html page

When clicks in the product ‘See More Details’ Button

Then the product.html page (with the Product ID in the URL) is open.

Teste 1.4. Navigation ‘Cart’ Button

When clicks in the Navigation ‘Cart’ Button

Then the ShoppingCart.html page is displayed.

2. Product Page (product.html)

Teste 2.1. Product page URL contains product ID

When the product.html page is loaded

Then the product ID should be present in the page URL.

Teste 2.2. Fail AJAX Request

When the product.html page is loaded

And the AJAX request fails

Then user should receive an error – “Could not connect to the server! Please try again later.”

Teste 2.3. Success AJAX Request

When the product.html page is loaded

And the AJAX request is successful

Then the page should display the details (name, image, description, varnish and price) from the selected oak furniture.

Teste 2.4. Select a Varnish

When clicks in the drop-down list of available Varnish

Then all the available varnish for that product should be displayed and can be selected.

Teste 2.5. Add to cart Button – without varnish

When clicks in the “Add to Cart” button in the product.html page

And the Varnish was not selected

Then the user should receive an error message – “Please select a Varnish!” and border should turn red.

Teste 2.6. Add to cart Button – with varnish

When clicks in the “Add to Cart” button in the product.html page

And the Varnish was previously selected

Then the user should receive a message – Product name & “ successfully added to the cart!”

And a “Go to Cart” Button should also be displayed.

Teste 2.7. ‘Go to Cart’ Button

Given a user that already clicked in the “Add to Cart” button (with a valid varnish)

When the user clicks in the “Go to Cart” button

Then the shoppingCart.html page is loaded.

3. Cart Page (shoppingCart.html)

Teste 3.1. Empty Cart page load

Given a user without any product in the cart

When the shoppingCart.html page is loaded

Then the user should have the feedback that the cart has 0 items selected

Teste 3.2. Cart page load with at least one product

Given a user with at least one product in the cart

When the shoppingCart.html page is loaded

Then a list of the products (image, name, varnish, quantity and price) in the cart and total amount (products amount, shipping and total amount) are displayed

Teste 3.3. Increase button “+”

Given a user with at least one product in the cart

When clicking in the increase button “+” from a product

Then the quantity of that product should increase in one unit

And the “Product(s) Amount” and “Total amount” should increase by the amount of the product unit price

Teste 3.4. Decrease button “-” with more than one unit

Given a user with at least one product in the cart

When clicking in the decrease button “-” from a product with more than one unit

Then the quantity of that product should decrease in one unit

And the “Product(s) Amount” and “Total amount” should be reduced by the amount of the product unit price

Teste 3.5. Decrease button “-” with only one unit

Given a user with at least one product in the cart

When clicking in the decrease button “-” from a product with only one unit

Then the product will be removed from the Cart summary.

And the “Product(s) Amount” and “Total amount” should decrease by the amount of the product unit price

Teste 3.6. Remove item link

Given a user with at least one product in the cart

When clicking in the “Remove item” link from a product

Then the product will be removed from the Cart summary.

And the “Product(s) Amount” and “Total amount” should decrease by the amount that was associated to that product

Teste 3.7. Checkout Button – without any item in the cart

Given a user without any product in the cart

When clicking in the “Checkout” button

Then the user should receive an error message - “Error - Your Cart is empty!”.

Teste 3.8. Checkout Button – with at least one item in the cart

Given a user with at least one product in the cart

When clicking in the “Checkout” button

Then a user Form should be displayed with the following fields: First name, Last name, Email, Address and City.

Teste 3.9. User input First name empty

When clicking in the user input form “Submit” button

And the First name field is empty

Then the user should receive a message – “Insert your name” and field border should turn red (function ‘emptyControl ()’ > 0, function ‘inputValidation()’ = false)

Teste 3.10. User input Last name empty

When clicking in the user input form “Submit” button

And the Last name field is empty

Then the user should receive a message – “Insert your Last name” and field border should turn red (function ‘emptyControl ()’ > 0, function ‘inputValidation()’ = false)

Teste 3.11. User input E-mail empty

When clicking in the user input form “Submit” button

And the E-mail field is empty

Then the user should receive a message – “Insert your E-mail” and field border should turn red (function ‘emptyControl ()’ > 0, function ‘inputValidation()’ = false)

Teste 3.12. User input Address empty

When clicking in the user input form “Submit” button

And the Address field is empty

Then the user should receive a message – “Insert your Address” and field border should turn red (function ‘emptyControl ()’ > 0, function ‘inputValidation()’ = false)

Teste 3.13. User input City empty

When clicking in the user input form “Submit” button

And the City field is empty

Then the user should receive a message – “Insert your City” and field border should turn red (function ‘emptyControl ()’ > 0, function ‘inputValidation()’ = false)

Teste 3.14. User input invalid E-mail format

When clicking in the user input form “Submit” button

And the E-mail field is not empty but does not have a valid E-mail format (ex: joe@email.com)

Then a user should receive a message – “The E-mail format is invalid” and field border should turn red” (function ‘controlEmail ()’ > 0, function ‘inputValidation ()’ = false)

Teste 3.15. User input invalid E-mail format

When clicking in the user input form “Submit” button

And the field in the user input form is filled (and has a valid format for the E-mail field)

Then the field border should turn green (function ‘controlEmail ()’ = 0, function ‘emptyControl ()’ = 0, function ‘inputValidation ()’ = true)

Teste 3.16. Submit order button - Fail AJAX Request

Given a user with all fields in the user form correctly filled (function ‘inputValidation ()’ = true)

When clicking in the user input form “Submit” button

And the AJAX request fails

Then user should receive an error – “Could not connect to the server! Please try again later.”

Teste 3.17. Submit order button - Success AJAX Request

Given a user with all fields in the user form correctly filled (function ‘inputValidation ()’ = true)

When clicking in the user input form “Submit” button

And the AJAX request is successful

Then the orderPage.html page is loaded.

4. Order Confirmation Page (orderPage.html)

Teste 4.1. Order confirmation page load

Given a user

When the orderPage.html page is loaded

Then the user should have the following information displayed:

- the Order ID
- the list of products in the order
- the total amount of the purchase
- the user information
- the delivery destination information

Teste 4.2. Cart information reset

Given a user

When the orderPage.html page has finished loading

Then the cart data, order Id and client data are deleted from the local storage.