# Knowledge Evaluation for QE

Name: Ricardo Fernandes de Souza Date: 10/06/2025

* **The candidate should not have any device with internet access** (e.g. mobile phones) when executing the test.
* Write your answers in English.
* Write down assumptions whenever necessary for better understanding.

Good luck.

# 1) Cover letter

Write an introduction letter (cover letter) in English describing your goals and pointing out your strongest abilities. Keep it simple and objective.

***[RICARDO]Dear team,***

***My name is Ricardo Fernandes de Souza, and I specialize in testing and defect management for critical systems in the telecommunications sector. I have solid experience in Quality Assurance, working with both manual and automated testing to ensure the reliability and performance of systems under high-demand conditions.***

***I was appointed Scrum Master of the automation squad within the support team, a role that strengthened my leadership, collaboration, and continuous improvement skills. I focus on designing test solutions, optimizing workflows, and identifying defects early—before they impact the end-user. I also contribute to organizing and streamlining team processes to ensure timely and effective deliveries.***

***I believe my background in critical systems can add significant value to RDI by helping create more agile and efficient QA processes. With an analytical mindset and results-driven approach, I aim to enhance product quality, refine validation strategies, and foster innovation. I’m excited about the opportunity to contribute to your QA team and make a meaningful impact.***

***Thank you for your consideration, looking forward to work with you.***

***Sincerely,***

***Ricardo Fernandes de Souza***

# 2) Please, pay attention to the following system components of a restaurant software solution.

# Point of Sale (POS)

It handles the order taking and tendering processes. The application user interface shows all screens with products and specialty function buttons. These transactions are sent to Kitchen Monitors so that they can be managed from the kitchen and expedition sides.

**Kitchen Monitors**

Monitors display several different colors, images, and icons that help the crew visualize the order and assembly of products. These system changes increase the speed of service and employee productivity.

**Customer ordering flow**

POS -> Kitchen Monitors

**Scenario:**

"Customer orders a meal. Crew registers meal. It is prepared and delivered to the customer"

Use the system components description to answer question-based on given scenario.

**Please, create a BDD based on the scenario given above.**

***[RICARDO] BDD (Behavior-Driven Development) – Using Gherkin format:***

***Feature: Meal Ordering and Delivery Flow  
 Scenario: Customer orders a meal and receives it  
 Given the POS system is ready for a new order  
 When the crew registers a customer’s meal  
 Then the Kitchen Monitor should display the meal details  
 And the meal should be prepared  
 And the customer should receive the correct meal***

**Please, specify what should be validated on automation.**

**[RICARDO]** ***As a general rule, the ideal topics for automation are processes that do not belong to unstable features or flows that are likely to change unpredictably. Therefore, the foundation of automation must start from stable and repetitive patterns. Based on that, here are some points that I considered suitable for automation in this scenario:***

* ***POS correctly registers orders***
  + ***Because it is the entry point of the ordering process and critical for ensuring the order is captured with the right data. If the POS fails, the entire chain (kitchen, delivery, customer experience) is affected.***
* ***Kitchen Monitor receives and displays the correct order***
  + ***Because it validates communication between modules. Ensuring that the kitchen staff receives the right information is essential for operational efficiency and customer satisfaction.***
* ***Order status progresses from “received” → “in preparation” → “ready”***
  + ***Because this reflects the business logic and process flow. Validating these transitions through automation helps guarantee that internal systems reflect the real status of an order.***
* ***End-to-end track of order from POS to delivery***
  + ***Because this ensures that the full process works as expected, and it validates that no data is lost or misrouted during the transitions. It’s crucial for detecting integration or communication issues.***
* ***Response time between order and kitchen update***
  + ***Because performance is a key quality attribute. Delays in kitchen updates can directly affect service time, especially in high-demand environments. Speaking of McDonald's, where speed is a key requirement for service, performance, volume, and quality must be prioritized***

**Write automation pseudo-code.**

**[RICARDO]** ***I will be using pseudocode in Cypress, which is the automation tool I have the most familiarity with***

describe('Meal Ordering and Delivery Flow', () => {

 it('Customer orders a meal and receives it', () => {

        // Given the POS system is ready for a new order

cy.visit('/pos'); // Access the POS interface

cy.**get**('#pos-status').should('contain', 'Ready'); // Ensure the system is ready

// When the crew registers a customer’s meal

        cy.**get**('#menu-item-burger').click(); // Select "Burger"

       cy.**get**('#add-to-order').click(); // Add it to the order

       cy.**get**('#submit-order').click(); // Submit

       cy.**get**('.order-confirmation').should('exist'); // Confirm order

       // Then the Kitchen Monitor should display the meal details

       cy.visit('/kitchen-monitor'); // Navigate to kitchen’s monitor

       cy.contains('Burger').should('be.visible'); // Check that the order appears

       cy.**get**('.order-status').should('contain', 'Received'); // Check status

       // And the meal should be prepared

        cy.**get**('.start-preparation-btn').click(); // Start meal preparation

        cy.**get**('.order-status').should('contain', 'In Preparation');

      // And the customer should receive the correct meal

      cy.**get**('.mark-ready-btn').click(); // Meal ready is marked

      cy.visit('/pos');

      cy.**get**('.deliver-order-btn').click(); // Confirm delivery

      cy.**get**('.order-status').should('contain', 'Delivered'); // Validate final status

    });

});

**If it was not possible to automate a component using any available test tool, how would you proceed?  
[RICARDO]** ***If it is not possible to automate the tests, then it will be necessary to perform structured and properly documented manual validations for them:***

* ***Validate API/backend responses that update the UI.***
* ***Apply exploratory testing.***
* ***Simulators to mock unavailable systems***

# 3) Automation test

Design relevant test scenarios to ensure the proper functionality of a login page. Develop automated tests based on these scenarios and upload the project to a Git repository.

**Target URL:**  
<https://the-internet.herokuapp.com/login>

**[RICARDO]** ***Please find the automated test project at the following repository:***

<https://github.com/RicardoFernandesSouza/RDI-QA-TEST.git>