

Proyecto QA Integral

1. Información General

Nombre: Alejandro Franco Acosta

Rol: QA Tester

Periodo: Sprint 1 – Sprint 2

Herramientas:

- Jira/Xray
- Python
- Selenium
- Pytest
- JUnit
- XML
- MongoDB
- Mongo Compass
- SoapUI
- Git y GitHub
- Docker
- DOM (Document Object Model)

2. Introducción

El presente informe documenta el trabajo realizado durante el Proyecto QA Integral, cuyo objetivo es demostrar mis conocimientos aplicados al aseguramiento de calidad sobre distintas capas de un sistema, combinando pruebas manuales y automatizadas. El proyecto abarcó validaciones a nivel de API, Base de Datos y UI, manteniendo trazabilidad completa mediante Jira/Xray que es nuestra herramienta de gestión clave para mantener el orden y la planeación.

3. Alcance del Proyecto

Las pruebas realizadas cubren las siguientes áreas:

- **API:** Validación de servicios SOAP mediante WSDL.
- **Base de Datos:** Validaciones sobre MongoDB (estructura, consistencia y reglas básicas).
- **UI:** Automatización del flujo de login en aplicación web.
- **Gestión QA:** Registro, ejecución y trazabilidad de pruebas en Jira/Xray.

Fuera de alcance:

- Pruebas de performance
- Pruebas de seguridad
- Integración CI/CD

4. Pruebas de API (SOAP)

4.1 Descripción

Se validó un servicio SOAP público utilizando SoapUI. A partir del WSDL se generaron operaciones y se crearon Test Suites y Test Cases para validar:

- Disponibilidad del servicio
- Código de respuesta HTTP
- Estructura del mensaje SOAP
- Resultado esperado de la operación

4.2 Evidencias

- Request y Response de API en SoapUI (open source)

The screenshot shows the SoapUI interface with the following details:

- Toolbar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Endpoint Explorer:** Shows a project named "QA_SOAP_CALCULATOR" containing "CalculatorSoap" with methods: Add, Divide, Multiply, Subtract. It also lists "CalculatorSoap12" and "TestSuite 001".
- Request View:** A "Request 1" window for "http://www.dneonline.com/calculator.asmx". The "Raw" tab displays the SOAP request XML:


```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header/>
<soapenv:Body>
<divide>
<tem:intA>15</tem:intA>
<tem:intB>5</tem:intB>
</divide>
</soapenv:Body>
</soapenv:Envelope>
```
- Response View:** A "Response 1" window showing the SOAP response XML:


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
<divideResponse xmlns="http://tempuri.org/">
<divideResult>3</divideResult>
</divideResponse>
</soap:Body>
</soap:Envelope>
```
- Request Properties:** A table showing properties for "Request 1":

Property	Value
Name	Request 1
Description	
Message Size	298
Encoding	UTF-8
Endpoint	http://www.dneon...
- Bottom Navigation:** SoapUI log, http log, jetty log, error log, wsm log, memory log.

- Test Suite, Test Case y Assertions(contains)

The screenshot shows the SoapUI interface with the following details:

- Toolbar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Endpoint Explorer:** Shows a project named "QA_SOAP_CALCULATOR" containing "CalculatorSoap" with methods: Add, Divide, Multiply, Subtract. It also lists "CalculatorSoap12" and "TestSuite 003".
- Test Case View:** A "TestCase 003" window for "SOAP Request 003 - División". The "Raw" tab displays the SOAP request XML:


```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header/>
<soap:Body>
<divide>
<tem:intA>25</tem:intA>
<tem:intB>5</tem:intB>
</divide>
</soap:Body>
</soap:Envelope>
```
- Response View:** A "Response 003" window showing the SOAP response XML:


```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
<divideResponse xmlns="http://tempuri.org/">
<divideResult>5</divideResult>
</divideResponse>
</soap:Body>
</soap:Envelope>
```
- Assertions:** A status bar at the bottom indicates "SOAP Response - VALID" and "Valid HTTP Status Codes - VALID".
- Bottom Navigation:** Assertions (2), Request Log (3).

- Se registra evidencias en Jira

1.2 Pruebas de Servicios SOAP

Description
Proyecto GitHub - Alex-af-2024/proyecto-qa-integral

Subtasks

Work	Pri...	Stor...	As...	Status
QA2-9 1.2.1 Importar un WSDL en SoapUI	= M	AF	DONE	
QA2-10 1.2.2 Crear test suite + test case	= M	AF	DONE	
QA2-11 1.2.3 Crear assertions(status, XML output, reglas)	= M	AF	DONE	
QA2-12 1.2.4 Exportar evidencias (XML + capturas)	= M	AF	DONE	
QA2-13 1.2.5 Registrar caso de prueba manual en Xray	= M	AF	DONE	

Linked work items
Add linked work item

Details

- Assignee: alejandro franco
- Reporter: alejandro franco
- Development: Create branch, Create commit
- Labels: None
- Due date: Dec 16, 2025
- Start date: Dec 15, 2025
- Sprint: None +1
- Priority: Medium
- Parent: QA2-11 Proyecto Integral del

4.3 Conocimientos aplicados

- Lectura e interpretación de WSDL
- Estructura de mensajes SOAP
- Uso de assertions funcionales
- Organización de pruebas en suites y casos

4.4 Recomendaciones

- Versionar los proyectos SoapUI en el repositorio
- Separar datos de prueba cuando el servicio lo permita
- Documentar claramente los servicios probados

5. Pruebas de Base de Datos (MongoDB)

Se sugiere usar Docker mongo para mantener un entorno de pruebas limpio en el ordenador y hacer el proyecto portable.

5.1 Descripción

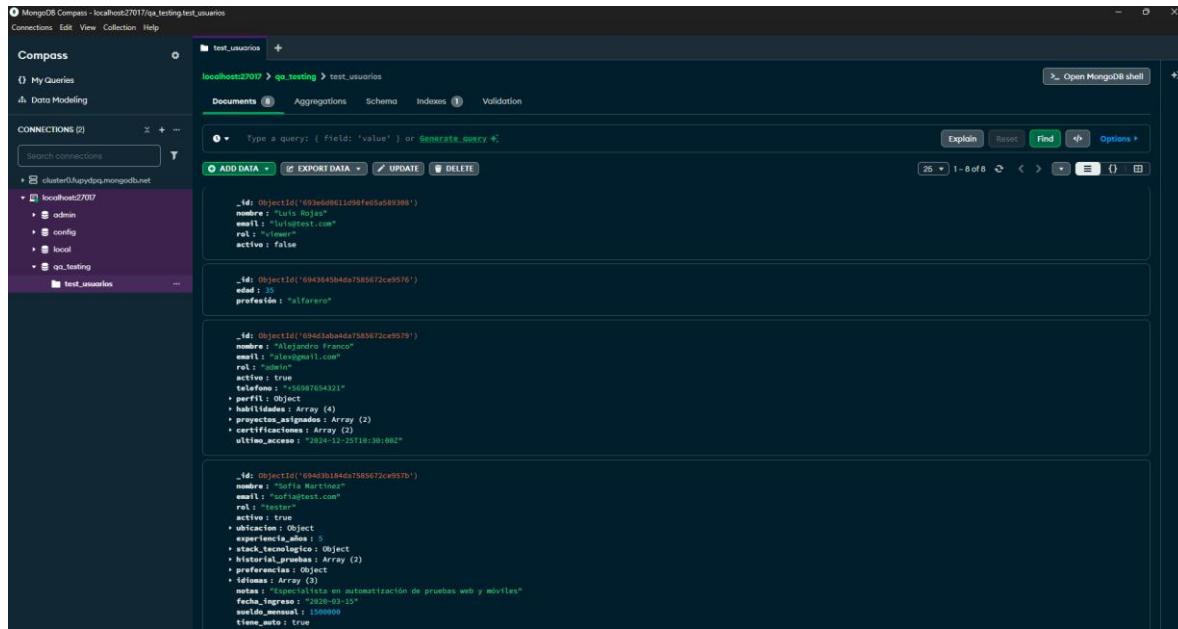
Se creó una base de datos de prueba en MongoDB para validar reglas básicas de integridad y consistencia de datos. Las validaciones se realizaron tanto de forma manual (MongoDB Compass) como automatizada mediante scripts en Python usando PyMongo.

Las validaciones incluyeron:

- Existencia de registros
- Validación de usuarios activos
- Validación de formato de emails
- Verificación de campos obligatorios

5.2 Evidencias

- Creación de base de datos local con MongoDB usando Mongo Compass como visualizador.
- Esta base de datos aprovecha el modelado flexible de documentos NoSQL.



The screenshot shows the MongoDB Compass interface connected to a local host at port 27017. The database 'qa_testing' is selected, and the collection 'test_usuarios' is displayed. The interface includes a left sidebar for connections and a main area for document management. The current view shows three documents in the list:

- Document 1:

```
_id: ObjectId('694d3ab4da7a85672ce9579')
nombre : "Luis Rojas"
correo : "luis@luis.com"
rol : "viewer"
activo : false
```
- Document 2:

```
_id: ObjectId('694d3ab4da7a85672ce9576')
edad : 35
profesion : "alfarero"
```
- Document 3:

```
_id: ObjectId('694d3ab4da7a85672ce9579')
nombre : "Alejandro Franco"
correo : "ale@gmail.com"
rol : "viewer"
activo : true
telefono : "+56987654321"
• habilidades : Array (4)
• proyectos_asignados : Array (2)
• certificaciones : Array (2)
ultimo_acceso : "2024-12-25T10:30:06Z"
```
- Document 4:

```
_id: ObjectId('694d3ab4da7a85672ce957b')
nombre : "Sofia Martinez"
correo : "sofia@sofia.com"
rol : "tester"
activo : true
ubicacion : Object
• latitud : 34.0
• longitud : -58.4
stack_tecnologico : Object
• historial_pruebas : Array (2)
• preferencias : Object
• intereses : Object
notas : "Especialista en automatización de pruebas web y móviles"
fecha_ingreso : "2020-03-15"
sueldo_mensual : 1500000
tiene_auto : true
```

- Ejecución de script para confirmar la existencia de usuarios activos en DDBB

The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** proyecto-qa-integral
- Explorer:** Shows the project structure under 'PROYECTO-QA-INTEGRAL':
 - __pycache__
 - .pytest_cache
 - api-test
 - db\mongo
 - validar_consistencia_usuarios.py
 - validar_usuarios.py
 - evidencias\screenshots
 - reportes\junit
 - ui-tests\tests
 - __pycache__
 - .pytest_cache
 - conftest.py
 - reportLogin.xml
 - test_login_selenium.py (marked as modified)
 - .gitignore
 - README.md
 - requirements.txt
 - test_browser.py
 - test.selenium-setup.py
- Code Editor:** Displays a Python script named 'validar_usuarios.py' with the following content:


```

db > mongo > validar_usuarios.py ...
1   from pymongo import MongoClient
2
3   # Conexión
4   client = MongoClient("mongodb://localhost:27017")
5
6   # Base de datos y colección
7   db = client["qa_testing"]
8   collection = db["test_usuarios"]
9
10  print("● Iniciando validaciones MongoDB...\n")
11
12  # Validación 1: existen usuarios activos
13  active_users = list(collection.find({"activo": True}))
14
15  if len(active_users) > 0:
16      print(f"✓ Existen usuarios activos: {len(active_users)} encontrado(s)")
17  else:
18      print("⚠ No existen usuarios activos")
19
20  print("\n✓ Validación MongoDB finalizada")
21
      
```
- Terminal:** Shows the command run in a virtual environment (venv):


```
(venv) PS C:\QA Tester\proyecto-qa-integral> python .\db\mongo\validar_usuarios.py
● Iniciando validaciones MongoDB...
✓ Existen usuarios activos: 3 encontrado(s)
✓ Validación MongoDB finalizada
○ (venv) PS C:\QA Tester\proyecto-qa-integral>
```

5.3 Conocimientos aplicados

- Uso de MongoDB Compass
- Modelado flexible de documentos.
- Conexión a MongoDB con PyMongo
- Validaciones lógicas mediante scripts

5.4 Recomendaciones

- Utilizar MongoDB en Docker para mayor portabilidad del proyecto
- Definir validaciones de esquema cuando el proyecto crezca
- Separar claramente datos de prueba y datos productivos

6. Pruebas UI Automatizadas (Selenium + Pytest)

6.1 Descripción

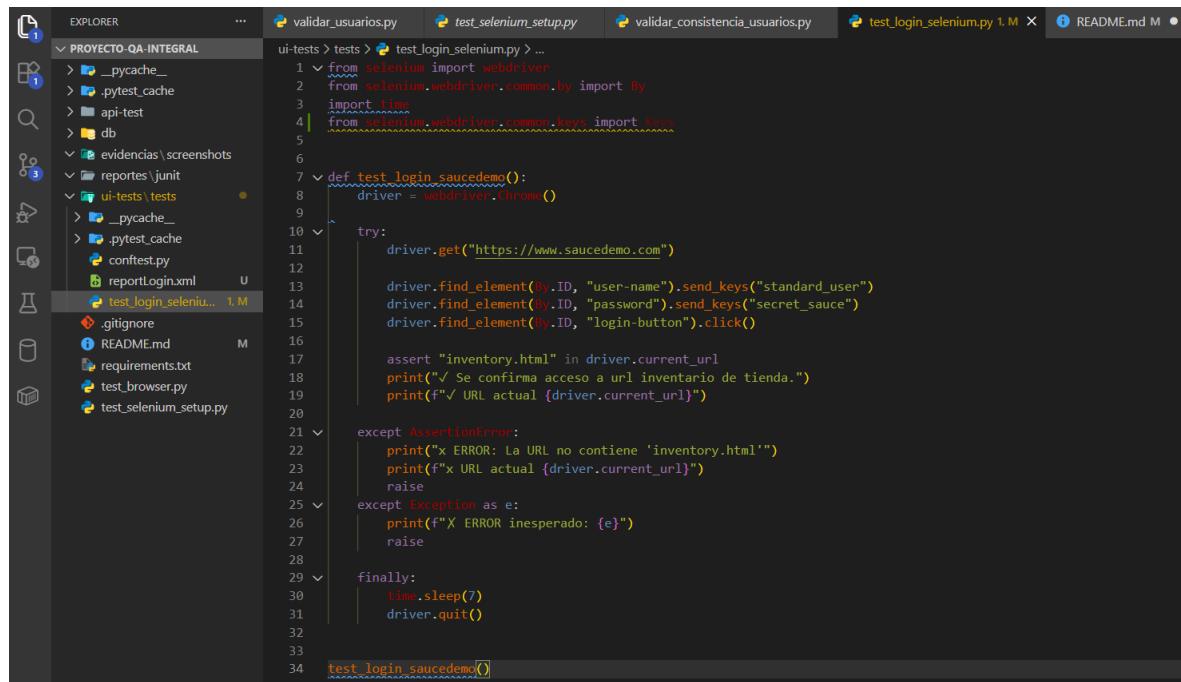
Se automatizó el flujo de login de una aplicación web utilizando Selenium WebDriver con Python y Pytest. El objetivo fue validar el acceso exitoso y comprobar la correcta redirección posterior al login.

El script incluye:

- Inicialización del navegador
- Interacción con elementos UI - DOM
- Aserciones funcionales
- Manejo básico de errores
- Captura de evidencias

6.2 Evidencias

- Script de automatización para login



The screenshot shows a code editor interface with several tabs at the top: 'validar_usuarios.py', 'test_selenium_setup.py', 'validar_consistencia_usuarios.py', 'test_login_selenium.py 1. M', and 'README.md'. The main pane displays a Python script for testing a login function. The script imports webdriver, By, time, and Keys from selenium. It defines a test function 'test_login_saucedemo()' that opens the saucedemo URL, inputs standard user credentials, and clicks the login button. It then asserts that the current URL contains 'inventory.html' and prints confirmation messages. If assertions fail, it prints error messages and raises exceptions. Finally, it sleeps for 7 seconds and quits the driver. The code is as follows:

```
1  from selenium import webdriver
2  from selenium.webdriver.common.by import By
3  import time
4  from selenium.webdriver.common.keys import Keys
5
6
7  def test_login_saucedemo():
8      driver = webdriver.Chrome()
9
10     try:
11         driver.get("https://www.saucedemo.com")
12
13         driver.find_element(By.ID, "user-name").send_keys("standard_user")
14         driver.find_element(By.ID, "password").send_keys("secret_sauce")
15         driver.find_element(By.ID, "login-button").click()
16
17         assert "inventory.html" in driver.current_url
18         print(f"✓ Se confirma acceso a url inventario de tienda.")
19         print(f"✓ URL actual {driver.current_url}")
20
21     except AssertionError:
22         print("✗ ERROR: La URL no contiene 'inventory.html'")
23         print(f"✗ URL actual {driver.current_url}")
24         raise
25     except Exception as e:
26         print(f"✗ ERROR inesperado: {e}")
27         raise
28
29     finally:
30         time.sleep(7)
31         driver.quit()
32
33
34 test_login_saucedemo()
```

- Captura de acceso a página con login ejecutado con script

Swag Labs

saucedemo.com/inventory.html

Un software automatizado de pruebas está controlando Chrome.

Swag Labs

Products

Sauce Labs Backpack
carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.



\$29.99

Add to cart

Sauce Labs Bike Light
A red light isn't the desired state in testing but it sure helps when riding your bike at night. Water-resistant with 3 lighting modes, 1 AAA battery included.



\$9.99

Add to cart

Sauce Labs Bolt T-Shirt
Get your testing superhero on with the Sauce Labs bolt T-shirt. From American Apparel, 100% ringspun combed cotton, heather gray with red bolt.



\$15.99

Add to cart

- Resultado usando Pytest en línea de comando

```
(venv) PS C:\QA Tester\proyecto-qa-integral> pytest .\ui-tests\tests\test_login_selenium.py -v
=====
platform win32 -- Python 3.13.3, pytest-9.0.2, pluggy-1.6.0 -- C:\venv_qa\venv\Scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\QA Tester\proyecto-qa-integral
collected 1 item

ui-tests/tests/test_login_selenium.py::test_login_saucedemo PASSED [100%]

===== 1 passed in 22.92s =====
o (venv) PS C:\QA Tester\proyecto-qa-integral> []
```

6.3 Conocimientos aplicados

- Automatización UI con Selenium
- Uso de selectores - DOM
- Aserciones con Pytest
- Estructuración de pruebas automatizadas

6.4 Recomendaciones

- Implementar Page Object Model para mayor mantenibilidad
- Externalizar credenciales y datos sensibles
- Integrar ejecución en pipelines CI/CD

7. Gestión de Pruebas con Jira y Xray

7.1 Descripción

Se utilizó Jira con el complemento Xray para:

- Registrar casos de prueba manuales
- Registrar pruebas automatizadas
- Crear Test Plans y Test Executions
- Importar resultados automatizados vía XML (formato JUnit)
- Mantener trazabilidad entre tareas, tests y ejecuciones

7.2 Evidencias

Creación de Test con pasos detallados

The screenshot shows a test management interface with a test case titled "Souce Demo - Login exitoso - Usuario Válido". The test description states: "Probar login exitoso de saucedemo.com con credenciales válidas." Below the description is a screenshot of a browser window showing a login form with fields for Username and Password, and a "Login" button. The browser title is "Swag Labs". To the right of the test details is a sidebar with sections for "Details", "Assignee" (alejandro franco), "Reporter" (alejandro franco), "Development" (Create branch, Create commit), "Labels" (None), "Due date" (None), "Start date" (None), "Priority" (Medium), and "Test Status" (Open Test Status). There is also a "More fields" section.

- Estructurar pasos de forma detallada y ordenada

The screenshot shows a test execution interface with a tab bar including "Test details", "Preconditions", "Test Sets", "Test Plans", "Test Runs", and "Test Repository". The "Test details" tab is active. The "Test Type" is set to "Manual". The main area displays a list of test steps numbered 1 to 5:

- 1 Abrir navegador
- 2 Ingresar username
- 3 Ingresar password
- 4 Click en login
- 5 Validar pantalla principal

Each step has a "Click to expand" button next to it. At the bottom of the interface are buttons for "Add Step", "New Step", and "Call Test".

- Se crea Test Execution y agrega Test Manual con sus pasos para su ejecución

The screenshot shows a Jira Test Management board titled 'QA2 board'. A specific test execution is selected, labeled 'Test: QA2-39'. The test details indicate it was passed ('PASSED') with a duration of 00:00:12. It was started and finished on December 26, 2025, at 12:38 PM. The assignee is unassigned, and it was executed by alejandro franco. The test version is v1, and the revision is also v1. The environment is 'Test Environments'. The 'Findings' section is collapsed. The 'Test details' section shows a test description: 'Probar login exitoso de saucedemo.com con credenciales válidas.'

- En la ejecución podemos registrar las evidencias según realicemos el paso a paso.

The screenshot shows the 'Steps' section of the test execution. It contains five steps, each with an action, data, expected result, and step state. Step 1: Action 'Abrir navegador', Data 'https://www.saucedemo.com/', Expected Result 'Página login visible', Step State PASSED. Step 2: Action 'Ingresar username', Data 'Usuarios Registrados: standard_user', Expected Result 'Username ingresado', Step State PASSED. Step 3: Action 'Ingresar password', Data 'Contraseña: secret_sauce', Expected Result 'Password ingresado', Step State PASSED. Step 4: Action 'Click en login', Data 'Datos de referencia: id="login-button"', Expected Result 'Usuario autenticado', Step State PASSED. Step 5: Action 'Validar pantalla principal', Data '• Dashboard visible
• URL contiene saucedemo.com/inventory.html
• Se visualiza listado de productos', Expected Result '• Dashboard visible
• URL contiene saucedemo.com/inventory.html
• Se visualiza listado de productos', Step State PASSED.

- En este caso tenemos PASSED para Test creado.

Se crea Test Execution y agrega XML de Test automatizado con Python

EXE - Automatizado - Login Exitoso

Description
Add a description...

Subtasks
Add subtask

Linked work items
Add linked work item

Tests

Overall Execution Status

1 PASSED TOTAL TESTS: 1

Rank	Key	Summary	Test Type	Dataset	#Defects	Status
1	QA2-45	test_Login_saucedemo	Generic	grid	0	PASSED

Total 1 issues

Details

- Assignee: alejandro franco
- Reporter: alejandro franco
- Development: Create branch
- Labels: None
- Due date: None
- Start date: None
- Priority: Medium
- Test Plans: Open Test Plans
- Test Environments: Open Test Environments

More fields

- Original estimate: 0m
- Time tracking: No time logged
- Components: Add components
- Sprint: None
- Team: None

7.3 Conocimientos aplicados

- Gestión de pruebas en Jira
- Uso de Xray para trazabilidad
- Importación de resultados automatizados (Documento estándar JUnit)
- Lectura e interpretación de reportes

7.4 Recomendaciones

- Definir convenciones de nombres para tests
- Centralizar evidencias en Test Executions
- Usar Test Plans para ciclos de prueba formales

8. Conclusiones

El Proyecto QA Integral permitió aplicar un enfoque completo de aseguramiento de calidad, abarcando múltiples capas del sistema y combinando pruebas manuales y automatizadas. Se adquirieron conocimientos prácticos en herramientas ampliamente utilizadas en la industria, así como en la gestión y trazabilidad de pruebas.

El proyecto deja una base sólida para evolucionar hacia prácticas más avanzadas como CI/CD, pruebas de performance y automatización a mayor escala.