



Cohorte 66 - Testing y Automatización de la alianza entre  
GOYN, MindHub y PROtalento.

## **Challenge Final - Informe**

### **Presentado por:**

Angel Vargas

Antonella Burbano

Erika Escalante

Yireth Rosero

Juan Camilo Cortes

Julieth Henao

Nelbis Olivero

Valentina Castrillon

Laura Castro

**NOVIEMBRE 2024**

## Contexto

El banco Ease tiene como objetivo desarrollar e implementar una aplicación web de Home Banking que ofrezca una experiencia de usuario ágil, eficiente y segura para realizar operaciones bancarias desde cualquier dispositivo.

Esta solución innovadora permitirá a los usuarios gestionar sus bienes y servicios financieros de manera rápida y conveniente, reduciendo la necesidad de acudir a una sucursal física y brindando acceso desde cualquier lugar, a través de una plataforma web responsive compatible con PC, teléfonos y tablets.

## Tecnologías usadas

Las tecnologías usadas en este proyecto fueron:

Herramientas de testing	Pruebas API
Zephyr squad	Postman
Herramientas de gestión	Herramientas de automatización
Jira	Katalon
Confluence	Selenium
Pruebas de performance	Creación de diagrama
Jmeter	Visual Paradigm.

## Metodologías usadas

### 1. Scrum:

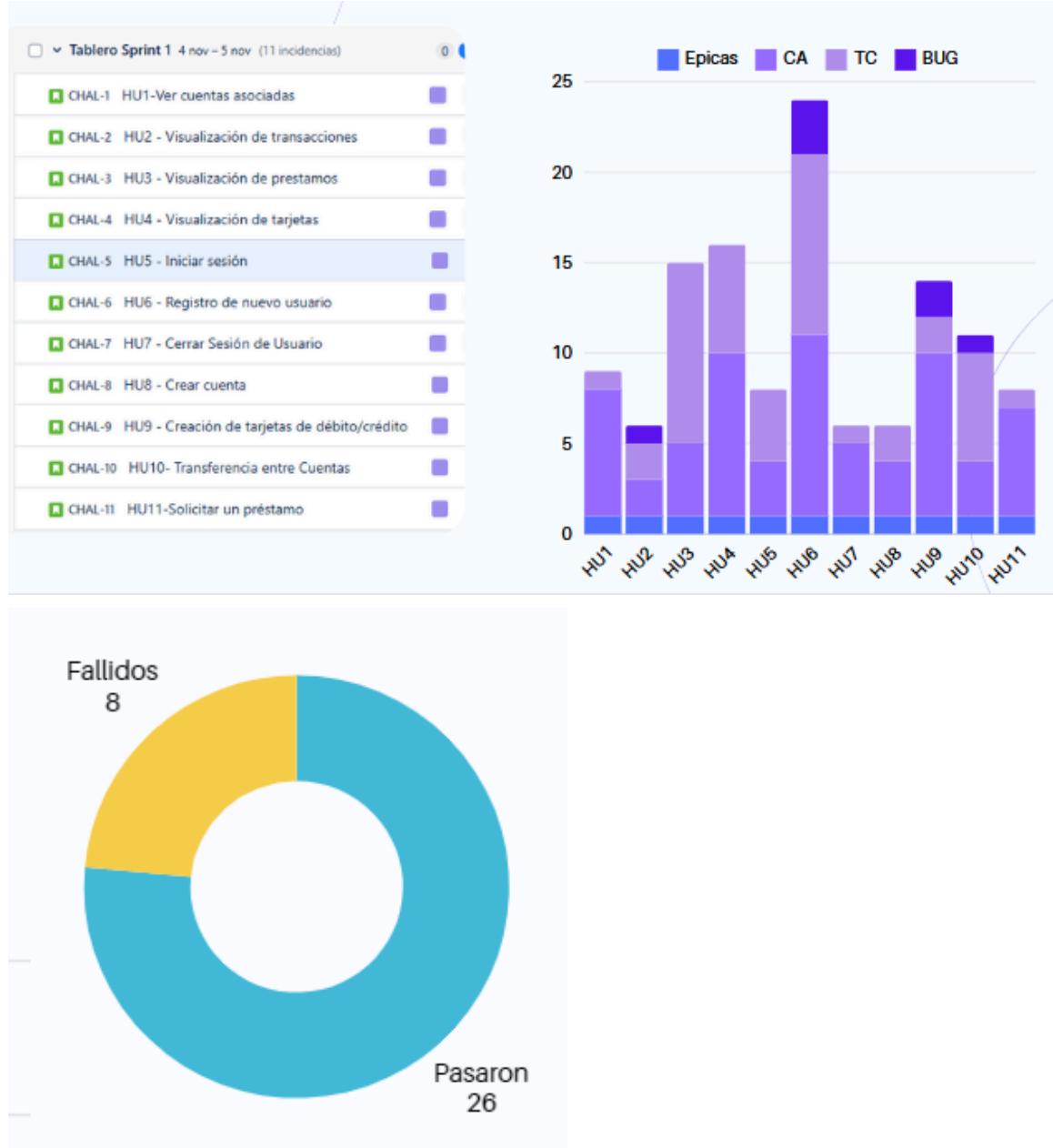
Es un marco de gestión de proyectos ágil que ayuda a los equipos a estructurar y gestionar su trabajo a través de un conjunto de valores, principios y prácticas.

### 2. Agile:

Es un enfoque o filosofía general para el desarrollo de software que se centra en la flexibilidad, la colaboración, la entrega iterativa y la respuesta rápida al cambio. Se basa en los principios y valores establecidos en el Manifiesto Ágil y promueve una mentalidad adaptativa y orientada al cliente.

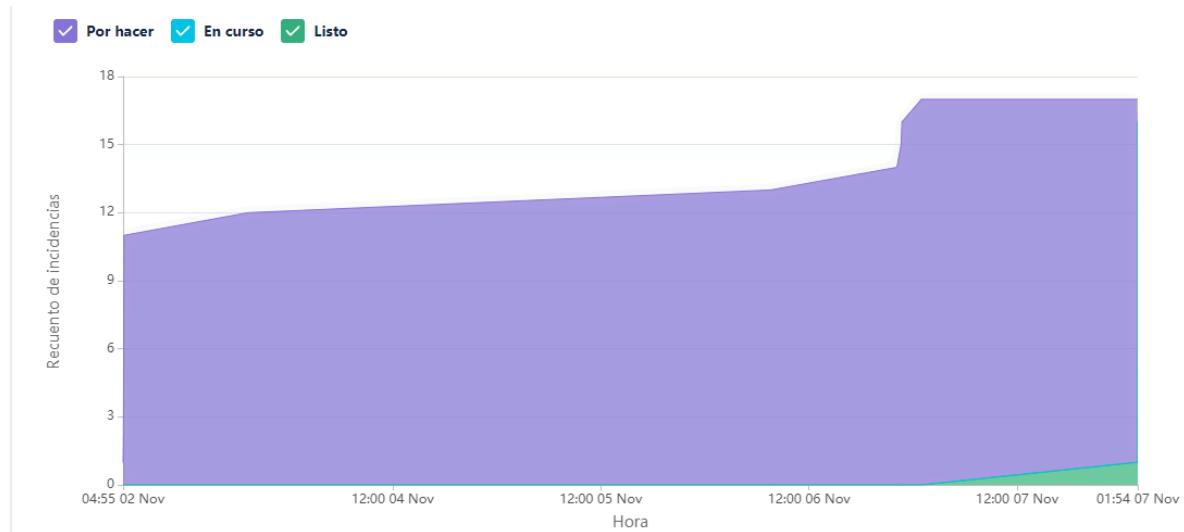
## GRÁFICOS

### 1.1 Historias de usuario, TC y Bugs

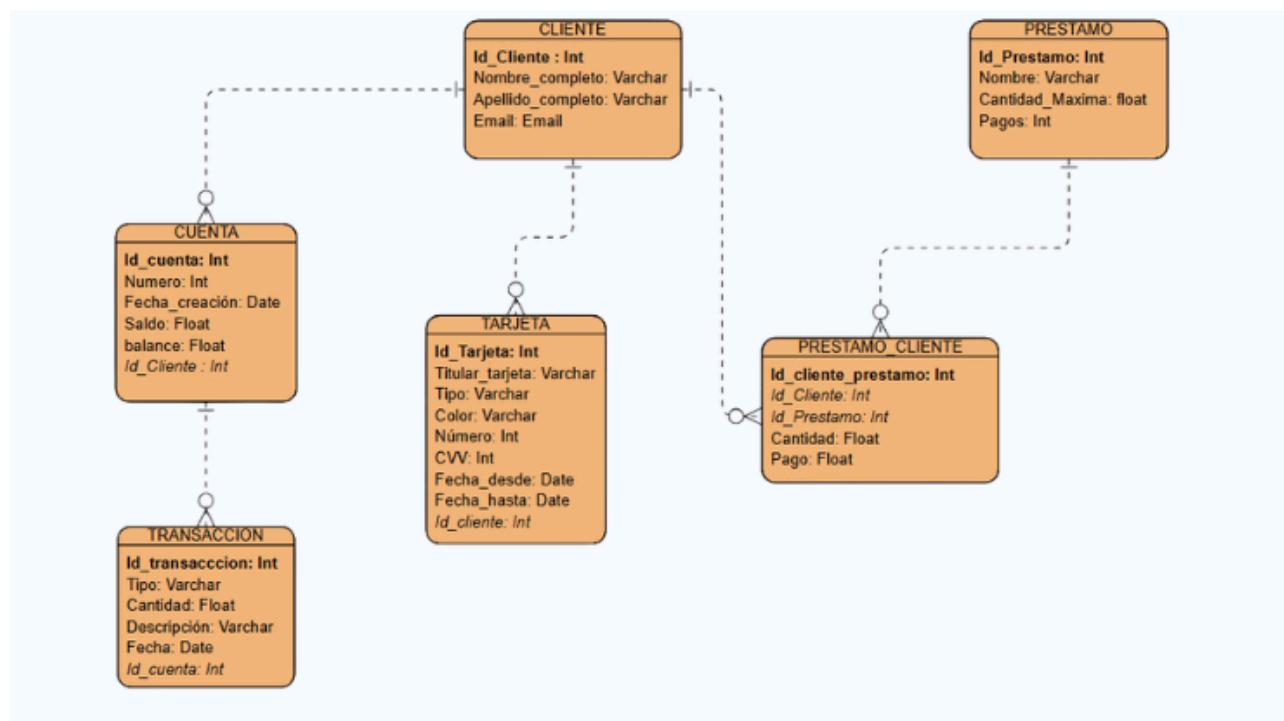


Obtuvimos 11 historias de usuario y 34 casos de prueba, los cuales en su ejecución obtuvimos 26 casos de pruebas que pasaron y 8 casos de prueba fallidos, de los cuales 3 fueron de severidad alta y 5 de severidad media.

## 1.2 Gráfico Cumulativo



## 1.3 Arquitectura de BD -- DIAGRAMA EDR



De acuerdo a los requerimientos pautados por los desarrolladores: un cliente tiene cuentas, con esta cuenta puede realizar transacciones. el cliente tiene tarjetas.

Optamos por crear una entidad llamada préstamo-cliente donde hay una relación de muchos a muchos con las entidades cliente y préstamos.

# Informe de Pruebas de Performance - JMeter

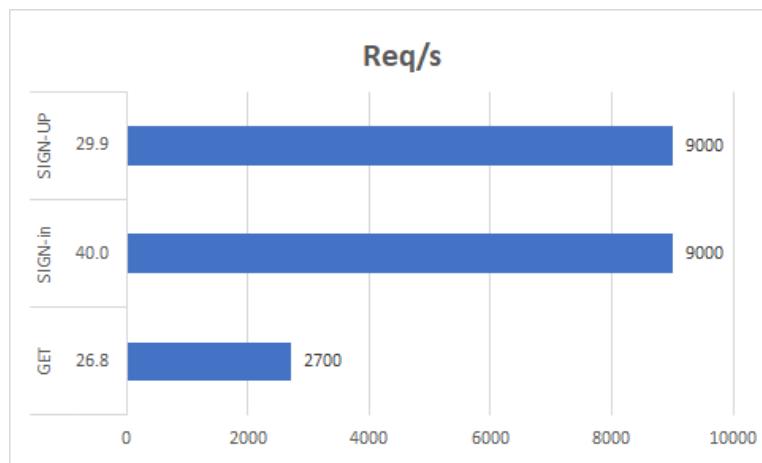
## 1. Resumen del Análisis:

Este informe analiza el rendimiento de diferentes tipos de solicitudes en términos de tiempos de respuesta, tasas de error y rendimiento en solicitudes por segundo. Los datos se presentan en gráficos para visualizar las métricas clave y comparar el desempeño de cada solicitud.

## 2. Gráficos de Análisis:

### Gráfico de Rendimiento (Req/s)

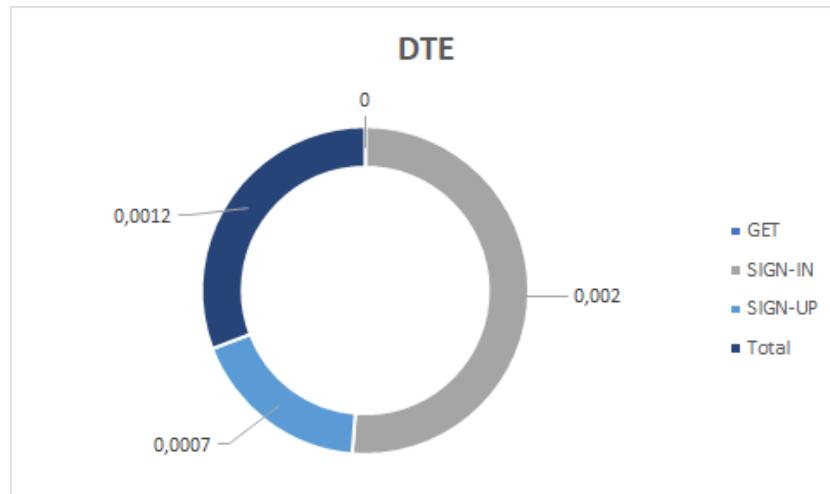
**Descripción:** Muestra la cantidad de solicitudes procesadas por segundo para cada tipo de solicitud (GET, SIGN-IN, SIGN-UP).



**Conclusión:** SIGN-UP demuestra la mejor capacidad de manejo de carga entre las solicitudes.

### Gráfico de Tasa de Error (DTE)

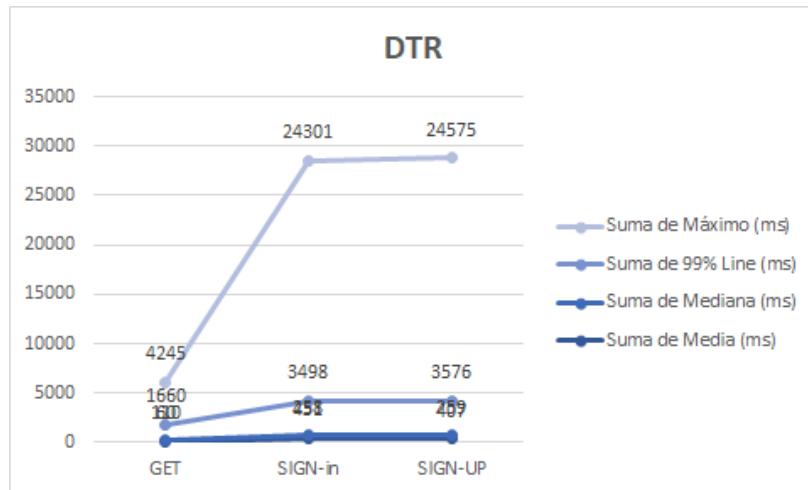
**Descripción:** Representa la proporción de errores en cada solicitud.



**Conclusión:** Aunque los errores son bajos en general, **SIGN-UP** muestra una menor confiabilidad en comparación con las demás solicitudes.

### Gráfico de Distribución de Tiempos de Respuesta (DTR)

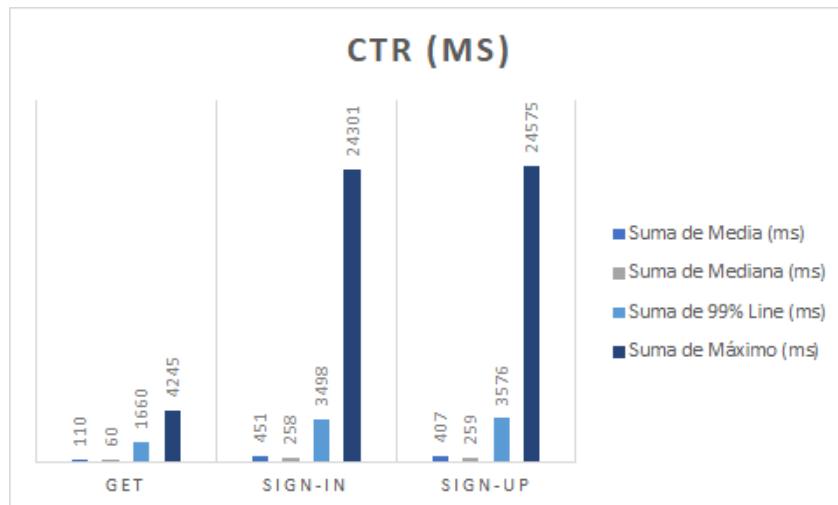
**Descripción:** Este gráfico de líneas visualiza la variación de tiempos de respuesta para cada solicitud, incluyendo el tiempo máximo, percentil 99%, mediana y media.



**Conclusión:** Las solicitudes tienen tiempos de respuesta medianos razonables, aunque los tiempos máximos son considerablemente altos, lo que sugiere posibles puntos de mejora en la estabilidad del sistema.

### Gráfico de Comparación de Tiempos de Respuesta (CTR)

**Descripción:** Este gráfico de columnas compara el tiempo medio, mediana, percentil 99% y tiempo máximo para cada solicitud.



**Conclusión:** La solicitud **GET** es la más eficiente en tiempos de respuesta, mientras que **SIGN-IN** y **SIGN-UP** muestran tiempos más altos, especialmente en los percentiles más elevados.

## **Conclusiones:**

1. **Rendimiento y Capacidad:** SIGN-UP tiene el mejor rendimiento en solicitudes por segundo, mientras que GET es la más rápida en tiempo de respuesta.
2. **Estabilidad:** La tasa de error es baja en general, pero SIGN-UP muestra un margen de mejora en términos de confiabilidad.
3. **Tiempos de Respuesta:** GET tiene un tiempo medio muy bajo, lo que indica alta eficiencia. SIGN-IN y SIGN-UP presentan tiempos máximos muy altos, sugiriendo picos de respuesta que pueden afectar la experiencia del usuario en momentos de alta demanda.

## **Recomendaciones:**

1. **Optimización de Tiempos Máximos:** Se recomienda analizar los tiempos de respuesta máximos en SIGN-IN y SIGN-UP para identificar posibles mejoras en la infraestructura o el código.
2. **Mejorar la Confiabilidad de SIGN-UP:** Con un 0.20% de tasa de error, SIGN-UP podría beneficiarse de un análisis de errores para mejorar la confiabilidad.

## Pruebas de API con Postman

Inicialmente, se crea el workspace en Postman y se configura una colección denominada "Home-Banking" para organizar las pruebas de la API.

Para cada request, se obtiene el CURL correspondiente a cada método HTTP desde la página web de la aplicación Home Banking. Esto permite importar los datos necesarios a Postman y facilita la ejecución de pruebas sistemáticas y precisas en cada endpoint.

### Métodos GET

- **Consulta de Transacciones Realizadas:**

La siguiente imagen muestra la respuesta obtenida al realizar una solicitud GET para consultar las transacciones realizadas por el usuario con sus respectivas cuentas. Esta solicitud recupera detalles completos de las operaciones efectuadas, proporcionando una vista detallada de su historial transaccional.

The screenshot shows the Postman interface with the 'Home-Banking' collection selected. In the left sidebar, under 'GET' requests, 'GET Transacciones realizadas' is highlighted with a red box. The main panel shows a 'GET' request to 'https://backend-java-homebanking.onrender.com/api/accounts/clients/current/accounts'. The 'Test Results' section displays a JSON response with a status of 200 OK, a response time of 520 ms, and a size of 1.04 KB. The JSON data is as follows:

```
[{"id": 6, "number": "VIN18150405", "creationDate": "2024-11-06T22:55:07.336009", "balance": 99880.0, "transactions": [{"id": 13, "transactionType": "CREDIT", "amount": 100000.0, "description": "Personal loan approved", "dateTransaction": "2024-11-06T23:13:01.840259"}]}
```

### Resultado:

```
[{"id": 6, "number": "VIN18150405", "creationDate": "2024-11-06T22:55:07.336009", "balance": 99880.0, "transactions": [{"id": 13, "transactionType": "CREDIT", "amount": 100000.0, "description": "Personal loan approved", "dateTransaction": "2024-11-06T23:13:01.840259"}]}
```

```

        },
        {
        "id": 14,
        "transactionType": "DEBIT",
        "amount": -50.0,
        "description": "Salary VIN88447798",
        "dateTransaction": "2024-11-06T23:14:59.839083"
    }
]
},
{
"id": 13,
"number": "VIN33602158",
"creationDate": "2024-11-06T23:26:59.244847",
"balance": 20000.0,
"transactions": [
{
        "id": 21,
        "transactionType": "CREDIT",
        "amount": 20000.0,
        "description": "Mortgage loan approved",
        "dateTransaction": "2024-11-06T23:37:05.840148"
    }
]
}
]

```

- Consulta de Información de Cuenta:**

El siguiente método GET permite obtener toda la información relacionada con las cuentas asociadas al usuario, proporcionando detalles completos sobre cada cuenta vinculada, incluido préstamos solicitados y tarjetas.

The screenshot shows the Postman interface with the following details:

- Collection:** Home-Banking
- Environment:** Home-Banking
- Request Method:** GET
- URL:** https://backend-java-homebanking.onrender.com/api/auth/current
- Headers:** (20)
- Body:** (Pretty, Raw, Preview, Visualize, JSON)
- Response:** 200 OK, 641 ms, 1.26 KB, JSON content displayed below

The response JSON content is as follows:

```

[
    {
        "id": 13,
        "transactionType": "CREDIT",
        "amount": 100000.0,
        "description": "Personal loan approved",
        "dateTransaction": "2024-11-06T23:13:01.840259"
    },
    {
        "id": 14,
        "transactionType": "DEBIT",
        "amount": -50.0,
        "description": "Salary VIN88447798",
        "dateTransaction": "2024-11-06T23:14:59.839083"
    }
]

```

## Resultado:

```

{
    "id": 4,
    "firstName": "Zoe",
    "lastName": "Hurtado",
    "email": "zoehur@gmail.com",
    "accounts": [
    {
        "id": 6,
        "number": "VIN18150405",
        "creationDate": "2024-11-06T23:26:59.244847",
        "balance": 20000.0,
        "transactions": [
        {
            "id": 21,
            "transactionType": "CREDIT",
            "amount": 20000.0,
            "description": "Mortgage loan approved",
            "dateTransaction": "2024-11-06T23:37:05.840148"
        }
        ]
    }
]
}

```

```
"creationDate": "2024-11-06T22:55:07.336009",
"balance": 99880.0,
"transactions": [
{
    "id": 13,
    "transactionType": "CREDIT",
    "amount": 100000.0,
    "description": "Personal loan approved",
    "dateTransaction": "2024-11-06T23:13:01.840259"
},
{
    "id": 14,
    "transactionType": "DEBIT",
    "amount": -50.0,
    "description": "Salary VIN88447798",
    "dateTransaction": "2024-11-06T23:14:59.839083"
}
],
{
    "id": 13,
    "number": "VIN33602158",
    "creationDate": "2024-11-06T23:26:59.244847",
    "balance": 20000.0,
    "transactions": [
{
        "id": 21,
        "transactionType": "CREDIT",
        "amount": 20000.0,
        "description": "Mortgage loan approved",
        "dateTransaction": "2024-11-06T23:37:05.840148"
}
]
},
{
    "id": 14,
    "number": "VIN28482529",
    "creationDate": "2024-11-06T23:27:23.940155",
    "balance": 20.0,
    "transactions": [
{
        "id": 20,
        "transactionType": "CREDIT",
        "amount": 20.0,
        "description": "Pago Juego VIN18150405",
        "dateTransaction": "2024-11-06T23:33:46.838539"
}
]
}
],
"loans": [
{
    "id": 7,
    "loanid": 1,
    "name": "Mortgage",
    "amount": 24000.0,
    "payments": 12
},
{
    "id": 5,
    "loanid": 2,
    "name": "Personal",
    "amount": 125000.0,
    "payments": 24
}
],
"cards": [
{
```

```

        "id": 5,
        "cardHolder": "Zoe Hurtado",
        "cardColor": "TITANIUM",
        "cardNumber": "5539-8780-9736-4374",
        "cvv": "617",
        "fromDate": "2024-11-06",
        "thruDate": "2029-11-06",
        "cardtype": "CREDIT"
    },
    {
        "id": 13,
        "cardHolder": "Zoe Hurtado",
        "cardColor": "GOLD",
        "cardNumber": "5760-3110-7975-7691",
        "cvv": "045",
        "fromDate": "2024-11-06",
        "thruDate": "2029-11-06",
        "cardtype": "CREDIT"
    }
]
}

```

- Consulta de Información de Tarjetas:**

El siguiente método GET recupera la información completa de las tarjetas asociadas al usuario, mostrando detalles relevantes sobre cada tarjeta vinculada.

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (Home-Banking), 'Environments', and 'History'. Under 'History', several API endpoints are listed: 'GET Transacciones realizadas', 'GET Información cuentas', 'GET Tarjetas' (which is highlighted with a red box), 'GET Créditos disponibles', 'POST Solicitud', 'POST Añadir tarjeta', 'POST Crear cuenta', 'POST Realizar transferencia', 'POST Realizar transferencia vacía', and 'POST Solicitar préstamo'. The main area shows a 'GET Tarjetas' request with the URL `https://backend-java-homebanking.onrender.com/api/clients/current/cards`. The 'Test Results' section displays a JSON response with two cards. The first card's details are highlighted with a red box in the 'Pretty' tab of the results table.

Key	Value	Description
id	5	Zoe Hurtado
cardHolder	Zoe Hurtado	TITANIUM
cardNumber	5539-8780-9736-4374	617
cvv	617	2024-11-06
fromDate	2024-11-06	2029-11-06
thruDate	2029-11-06	CREDIT

## Resultado:

```
[
{
    "id": 5,
    "cardHolder": "Zoe Hurtado",
    "cardColor": "TITANIUM",
    "cardNumber": "5539-8780-9736-4374",
    "cvv": "617",
    "fromDate": "2024-11-06",
    "thruDate": "2029-11-06",
    "cardtype": "CREDIT"
},
{
    "id": 13,
    "cardHolder": "Zoe Hurtado",
    "cardColor": "GOLD",
    "cardNumber": "5760-3110-7975-7691",
    "cvv": "045",
    "fromDate": "2024-11-06",
    "thruDate": "2029-11-06",
    "cardtype": "CREDIT"
}
]
```

```

        "thruDate": "2029-11-06",
        "cardtype": "CREDIT"
    },
    {
        "id": 13,
        "cardHolder": "Zoe Hurtado",
        "cardColor": "GOLD",
        "cardNumber": "5760-3110-7975-7691",
        "cvv": "045",
        "fromDate": "2024-11-06",
        "thruDate": "2029-11-06",
        "cardtype": "CREDIT"
    }
]

```

- **Consulta de Créditos Preprobados:**

El siguiente método GET recupera información detallada sobre los créditos preprobados disponibles para el cliente, incluyendo montos, condiciones y estado actual. Estos créditos están listos para ser solicitados por el cliente en cualquier momento.

The screenshot shows the Postman interface with the following details:

- Collection:** Home-Banking
- Request URL:** https://backend-java-homebanking.onrender.com/api/loans
- Method:** GET
- Response Status:** 200 OK
- Response Body (Pretty Print):**

```

1 [
2   {
3     "id": 3,
4     "name": "Automotive",
5     "maxAmount": 300000.0,
6     "payments": [
7       6,
8       12,
9       24,
10      36
11    ]
12  }
13 ]

```

## Resultado:

```

[
{
    "id": 3,
    "name": "Automotive",
    "maxAmount": 300000.0,
    "payments": [
        6,
        12,

```

```

        24,
        36
    ]
}
]

```

## Métodos POST.

- **Realizar Transferencia:**

El siguiente método POST se utiliza para ejecutar una transferencia entre cuentas. En el Body de la solicitud se incluyen los datos necesarios para realizar la operación, tales como:

```
{
  "sourceAccount": "VIN18150405",
  "destinationAccount": "VIN28482529",
  "amount": 20,
  "description": "Pago Juego"
}
```

En este caso, la respuesta de la API es un código de estado 201 Created, indicando que la transferencia se ha realizado exitosamente.

The screenshot shows the Postman interface with the following details:

- Collection:** Home-Banking
- Request:**
  - Method:** POST
  - URL:** https://backend-java-homebanking.onrender.com/api/transactions
  - Body (raw):**

```
1 {"sourceAccount": "VIN18150405", "destinationAccount": "VIN28482529", "amount": 20, "description": "Pago Juego"}
```
- Response:**
  - Status: 201 Created
  - Body: Transaction completed successfully

- **Transferencia con cuenta de destino vacía:**

Este método POST se utiliza para realizar una transferencia entre cuentas. En esta prueba, la solicitud no incluye la cuenta de destino, enviando un Body con los siguientes datos:

```
{
  "sourceAccount": "VIN33602158",
  "destinationAccount": "",
```

```

    "amount":1,
    "description":"salary"
}

```

La API responde con un código de estado 403 Forbidden, indicando que la transferencia no puede completarse debido a que el campo de "destinationAccount" está vacío, lo cual es un requisito obligatorio.

The screenshot shows the Postman interface with the following details:

- Collection:** Home-Banking
- Request:**
  - Method:** POST
  - URL:** https://backend-java-homebanking.onrender.com/api/transactions
  - Body:** Raw JSON (highlighted in red)
- Response:**
  - Status:** 403 Forbidden
  - Message:** "The destination account field must not be empty"

- **Solicitud de Préstamo:**

Este método POST se utiliza para solicitar un préstamo preaprobado. En este caso, al intentar realizar la solicitud, la API responde con un código de estado 403 Forbidden, indicando que el préstamo ya ha sido solicitado anteriormente, y por lo tanto no es posible procesar una nueva solicitud para el mismo préstamo.

The screenshot shows the Postman interface with the following details:

- Collection:** Home-Banking
- Request:**
  - Method:** POST
  - URL:** https://backend-java-homebanking.onrender.com/api/loans
  - Body:** Raw JSON (highlighted in red)
- Response:**
  - Status:** 403 Forbidden
  - Message:** "1"

- **Crear Cuenta:**

Este método POST permite la creación de una nueva cuenta, generando los datos correspondientes para su uso en transferencias o pagos. Sin embargo, en este caso, la API responde con un código de estado 403 Forbidden, ya que el cliente ya ha alcanzado el límite máximo de 3 cuentas disponibles y no puede crear una cuenta adicional.

The screenshot shows the Postman interface with a collection named "Home-Banking". A POST request is selected with the URL <https://backend-java-homebanking.onrender.com/api/accounts/clients/current/accounts>. The response status is 403 Forbidden. The body of the response is empty.

- **Solicitar Tarjeta:**

Este método POST permite la creación o solicitud de una nueva tarjeta, en este caso una tarjeta de crédito Gold. Sin embargo, la API responde con un código de estado 403 Forbidden, ya que el cliente ya posee una tarjeta de este tipo, la cual fue creada previamente.

The screenshot shows the Postman interface with a collection named "Home-Banking". A POST request is selected with the URL <https://backend-java-homebanking.onrender.com/api/clients/current/cards>. The response status is 403 Forbidden. The body of the response contains the JSON object `{"cardType": "CREDIT", "cardMembership": "GOLD"}`.

## REPORTE AUTOMATIZACIÓN SELENIUM Y KATALON

Tester selenium: Yennifer Arce

Tester katalon: Erika Escalante

Datos de entrada:

- URL: <https://frontend-react-homebanking.onrender.com/>
- Cliente: Melba Liliana Lorenzo
- Email: melblili@hotmail.com
- Contraseña: Aguacate12

**Análisis de automatización:** Al realizar el respectivo smoke test al home banking, identificamos los siguientes TC como prospectos de automatización, se usaron las herramientas SELENIUM y KATALON para efectuar la tarea. Se ejecutaron los siguientes casos: Registro de nuevo cliente, inicio de sesión, contraseña incorrecta, creación automática de cuenta, creación de tarjeta crédito, creación de tarjeta débito, formulario de mis propias cuentas, datos de tarjeta, lista de cuenta, préstamos no existentes, préstamos activos y cierre de sesión. Todos fueron ejecutados con éxito.

The screenshot shows the Selenium IDE interface with the following details:

- Project:** CHALLENGE FINAL
- Tests:** A list of 14 test cases, all marked with a checkmark:
  - ✓ Registrar un nuevo cliente
  - ✓ Validar el proceso de llenado del formulario mis propias
  - ✓ Validar funcionalidad de cierre de sesión exitoso
  - ✓ Validar inicio de sesión
  - ✓ Validar la creación de una tarjeta de crédito
  - ✓ Validar la creación de una tarjeta de débito
  - ✓ Verificar creación automática de cuenta
  - ✓ Verificar datos de la tarjeta
  - ✓ Verificar información de la cuenta
  - ✓ Visualizar Préstamos Activos
- Log:** A table showing the execution of the last two steps:

Step	Command	Target	Value
13	click on css: border-[#BDBDBD] OK		
14	click on css: bg-[#32CD32] OK		

The log also includes the message: "Registrar un nuevo cliente completed successfully".

## Registro de nuevo cliente:

Selenium IDE - CHALLENGE FINAL \*

Project: CHALLENGE FINAL \*

Tests + D E D S O

Search tests:

✓ Registrar un nuevo cliente

✓ Validar el proceso de llenado del formulario mis propias  
✓ Validar funcionalidad de cierre de sesión exitoso  
✓ Validar inicio de sesión  
✓ Validar la creación de una tarjeta de crédito  
✓ Validar la creación de una tarjeta de débito  
✓ Verificar creación automática de cuenta  
✓ Verificar datos de la tarjeta  
✓ Verificar información de la cuenta  
✓ Visualizar Préstamos Activos

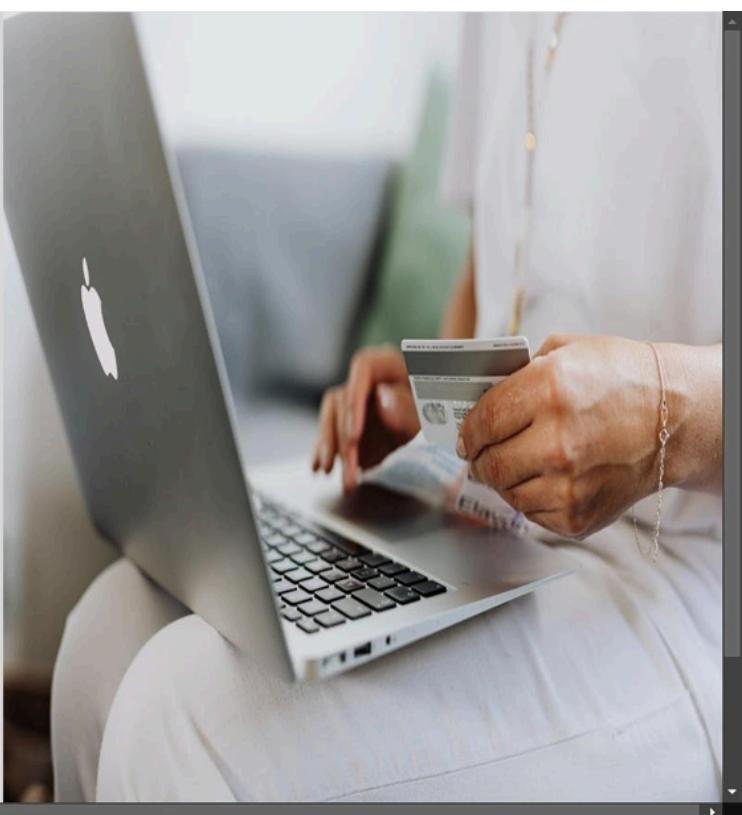
Command	Target	Value
1 ✓ open	/sign-in	
2 ✓ set window size	1552x840	
3 ✓ click	css= text-[#0066CC]	
4 ✓ click	id=first-name	Melba
5 ✓ type	id=first-name	Lorenzo
6 ✓ click	id=last-name	
7 ✓ type	id=last-name	lola@gmail.com
8 ✓ click	id=email	
9 ✓ type	id=password	melba12345.
10 ✓ click	id=password	melba12345.
11 ✓ type	id=password	
12 ✓ type	id=password	
13 ✓ click	css=border-[#B0B0B0]	

Command: open Target: /sign-in Value: Description:

Log Reference

13. click on css=border-[#B0B0B0] OK  
14. click on css=bg-[#32CD32] OK  
'Registrar un nuevo cliente' completed successfully

21:15:45  
21:15:48  
21:15:49



The image shows a woman's hands interacting with a silver laptop keyboard. She is holding a credit card in her right hand, likely entering payment information. The laptop screen displays a registration form for 'BANK EASE BANK SMART'. The form includes fields for First Name (Melba Liliana), Last Name (Lorenzo), Email (Melblili@hotmail.com), and Password (Aguacate12). A red error message at the bottom states 'The Email entered is already registered'. Below the form are 'Register' and 'Sign In' buttons.

## Inicio de sesión:

Selenium IDE - CHALLENGE FINAL \*

Project: CHALLENGE FINAL \*

Tests ▾

Search tests:

✓ Registrar un nuevo cliente  
✓ Validar el proceso de llenado del formulario mis propias  
✓ Validar funcionalidad de cierre de sesión exitoso  
**✓ Validar inicio de sesión**  
✓ Validar la creación de una tarjeta de crédito  
✓ Validar la creación de una tarjeta de débito  
✓ Verificar creación automática de cuenta  
✓ Verificar datos de la tarjeta  
✓ Verificar información de la cuenta  
✓ Visualizar Préstamos Activos

https://frontend-react-homebanking.onrender.com

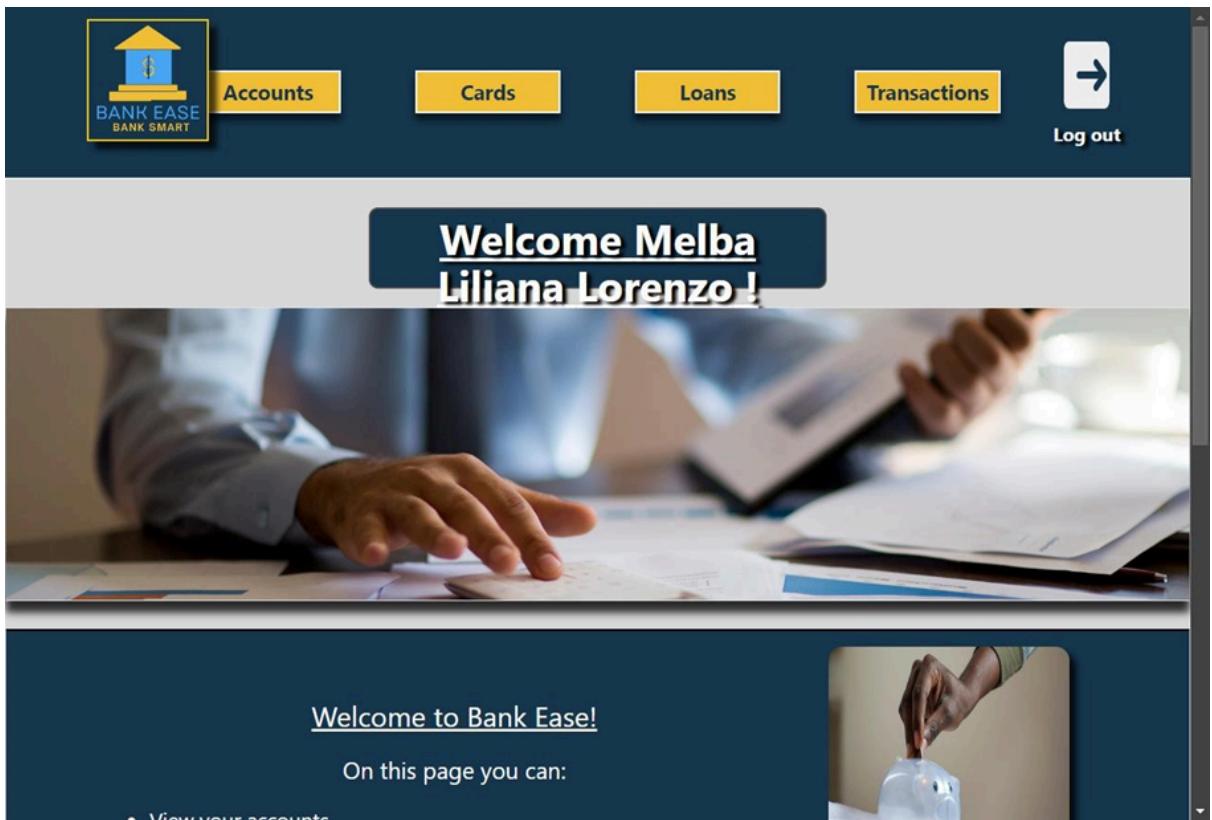
Command	Target	Value
1 open	/	
2 set window size	1074x824	
3 click	id=email	
4 type	id=password	melbalorenzo@gmail.com
5 click	id=password	MelbaL123
6 type	id=password	MelbaL123

Log Reference

Running: Validar la creación de una tarjeta de crédito'

1. open on / OK  
2. setWindowSize on 1552x840 OK  
3. click on linkText=Cards OK  
4. runScript on window.scrollTo(0,500) OK  
5. click on css=t\_w-[200px] OK  
6. mouseDown on id=cardType OK  
7. mouseUp on css=min-h-[calc(100vh-23vh)] OK  
8. click on css= min-h-[calc(100vh-23vh)] OK  
9. select on id=cardType with value label=CREDIT OK  
10. click on id=cardMembership OK

20.06.59  
20.07.00  
20.07.02  
20.07.04  
20.07.07  
20.07.10  
20.07.13  
20.07.16  
20.07.19  
20.07.22  
20.07.25



The screenshot shows the home banking interface for 'BANK EASE BANK SMART'. At the top, there's a navigation bar with a logo, 'Accounts', 'Cards', 'Loans', 'Transactions', a right-pointing arrow, and a 'Log out' button. Below the navigation is a large banner with the text 'Welcome Melba Liliana Lorenzo !' overlaid on a background image of hands working at a desk. The main content area has a dark blue header with the text 'Welcome to Bank Ease!' and 'On this page you can:'. A small image of a hand putting money into a piggy bank is visible on the right.

**Contraseña incorrecta:**



**BANK EASE**  
BANK SMART

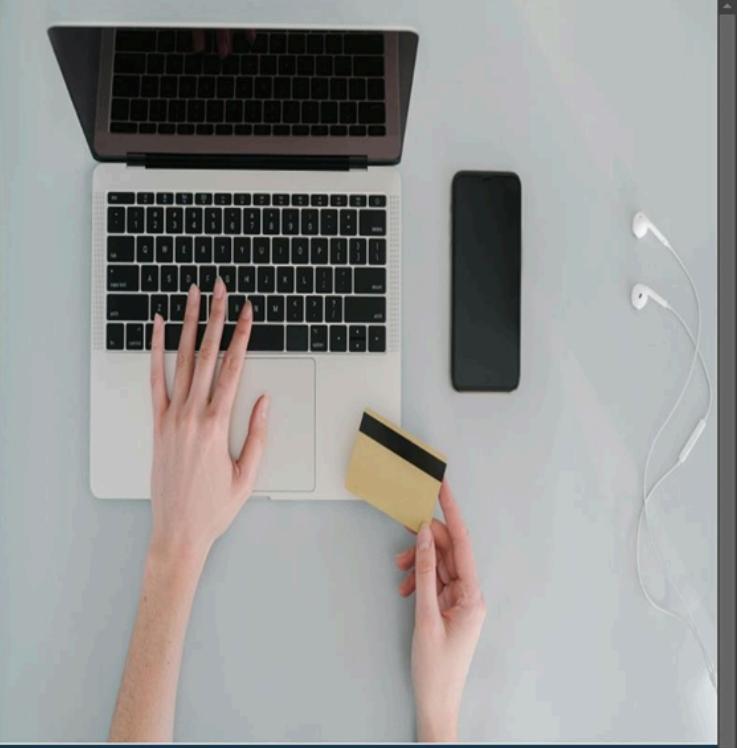
Email

Password

You have entered an incorrect email or password

[Sign in](#)   [Register](#)

© 2024 Bank Ease. All rights reserved.   Cohort 55 -William Ocanto



## Creación automática de cuenta:

Selenium IDE - CHALLENGE FINAL \*

Project: CHALLENGE FINAL \*

Tests: + D E S Q O

Search tests:

https://frontend-react-homebanking.onrender.com

Command	Target	Value
3 mouse over	linkText=Accounts	
4 click	linkText=Accounts	
5 mouse out	linkText=Accounts	
6 click	linkText=Accounts	
7 click	css=w-[15vw]	
8 click	css=bg-[M32CD32]	

Command: open / Target: / Value: Description:

Log Reference

Running: Validar la creación de una tarjeta de crédito'

1. open on / OK 20:06:59  
2. setWindowSize on 1552x840 OK 20:07:00  
3. click on linkText=Cards OK 20:07:02  
4. runScript on window.scrollTo(0,500) OK 20:07:04  
5. click on css:w-[200px] OK 20:07:07  
6. mouseDown on id=cardType OK 20:07:10  
7. mouseUp on css:min-height-[calc(100vh-23vh)] OK 20:07:13  
8. click on css:min-height-[calc(100vh-23vh)] OK 20:07:16  
9. select on id=cardType with value label=CREDIT OK 20:07:19  
10. click on id=cardMembership OK 20:07:22  
11. click on id=buttonSubmit OK 20:07:25

Welcome Melba  
Liliana Lorenzo !

Welcome to Bank Ease!

On this page you can:

- View your accounts
- Request a new account: Click the button to start the process (remember that you can have a maximum of 3 accounts)
- Access your transactions: Click on any account to see its movements.

Account number:  Account number:

## **Creación de tarjeta crédito:**

Selenium IDE - CHALLENGE FINAL \*

Project: CHALLENGE FINAL \*

Tests +

Search tests...

✓ Registrar un nuevo cliente  
✓ Validar el proceso de llenado del formulario mis propias  
✓ Validar funcionalidad de cierre de sesión exitoso  
✓ Validar inicio de sesión  
✓ Validar la creación de una tarjeta de crédito  
✓ Validar la creación de una tarjeta de débito  
✓ Verificar creación automática de cuenta  
✓ Verificar datos de la tarjeta  
✓ Verificar información de la cuenta  
✓ Visualizar Préstamos Activos

https://frontend-react-homebanking.onrender.com

	Command	Target	Value
9	select	id=cardType	label=CREDIT
10	click	id=cardMembership	
11	click	id=cardMembership	
12	select	id=cardMembership	label=SILVER
13	click	css=bg-[#F0F2CD32]	
14	click	css=px-4 nth-child(1)	

Command: open / #

Target: /

Value:

Description:

Log Reference

Running "Validar la creación de una tarjeta de crédito"

1. open on / OK 20:06:59  
2. setWindowSize on 1552x840 OK 20:07:00  
3. click on linkText=Cards OK 20:07:02  
4. runScript on window.scrollTo(0,500) OK 20:07:04  
5. click on css=w-[200px] OK 20:07:07  
6. mouseDown on id=cardType OK 20:07:10  
7. mouseUp on css=min-h-[calc(100vh-23vh)] OK 20:07:16  
8. click on css=min-h-[calc(100vh-23vh)] OK 20:07:19  
9. select on id=cardType with value label=CREDIT OK 20:07:22  
10. click on id=cardMembership OK 20:07:25

The image shows a screenshot of the Bank Ease mobile application. At the top, there is a dark blue header bar with the bank's logo on the left, which includes a stylized building icon and the text "BANK EASE" and "BANK SMART". To the right of the logo are four yellow rectangular buttons with black text: "Accounts", "Cards", "Loans", and "Transactions". Further to the right is a small white square icon with a black number "7" inside. On the far right of the header, the text "Log out" is visible. Below the header, the main content area has a light gray background. In the center, there is a large, rounded rectangular button with a dark gray gradient. Inside this button, the words "Apply for a Card" are written in a bold, white, sans-serif font, with "Card" being underlined. Below this button, the main body of the screen is a light gray box. On the left side of this box, there is a graphic illustration of a blue credit card with a yellow stripe and a yellow chip. The card is tilted and surrounded by three gold coins, with blue arrows pointing around it, suggesting motion or application. To the right of the graphic, there are two dropdown menus. The first menu is labeled "Select card type" and has "CREDIT" selected. The second menu is labeled "Select card membership" and has "SILVER" selected. At the bottom of the light gray box, a red text message reads "You already have a SILVER CREDIT card.". At the very bottom of the screen, there are two buttons: a green one on the left with the word "Apply" and a red one on the right with the word "Cancel".

## Creación tarjeta Débito:

Selenium IDE - CHALLENGE FINAL \*

Project: CHALLENGE FINAL \* [Edit](#)

Tests: [+ New](#) [Delete](#) [Copy](#) [Run](#) [Stop](#) [Help](#)

Search tests:  [Search](#)

https://frontend-react-homebanking.onrender.com

Command	Target	Value
20 click	id=cardMembership	
21 select	id=cardMembership	label=GOLD
22 click	css=bg-[#32CD32]	
23 mouse over	css=bg-[#32CD32]	
24 mouse out	css=gap-6 > border-2	
25 click	css=px-4 nth-child(1)	

Command:  [Open](#) [Close](#) [Copy](#) [Search](#)

Target:

Value:

Description:

Log Reference

Running: Validar la creación de una tarjeta de crédito

1. open on /OK  
2. setWindowSize on 1552x840 OK  
3. click on linkText:Cards OK  
4. runScript on window scrollTo(0,500) OK  
5. click on css:w-[200px] OK  
6. mouseDown on id=cardType OK  
7. mouseUp on css:min-h-[calc('100vh-23vh')] OK  
8. click on css:min-h-[calc('100vh-23vh')] OK  
9. select on id=cardType with value label=CREDIT OK  
10. click on id=cardMembership OK

20.06.59  
20.07.00  
20.07.02  
20.07.04  
20.07.07  
20.07.10  
20.07.13  
20.07.16  
20.07.19  
20.07.22  
20.07.25

BANK EASE BANK SMART [Log out](#)

Accounts Cards Loans Transactions

[Apply for a Card](#)

Select card type: DEBIT

Select card membership: TITANIUM

You already have a TITANIUM DEBIT card.

Apply Cancel

© 2024 Bank Ease. All rights reserved. Cohort 55 -William Ocanto [Instagram](#) [Facebook](#) [Messenger](#)

## Formulario de mis propias cuentas

Selenium IDE - CHALLENGE FINAL

Project: CHALLENGE FINAL\*

Tests +

Search tests...

✓ Registrar un nuevo cliente  
✓ Validar el proceso de llenado del formulario mis propias  
✓ Validar funcionalidad de cierre de sesión exitoso  
✓ Validar inicio de sesión  
✓ Validar la creación de una tarjeta de crédito  
✓ Validar la creación de una tarjeta de débito  
✓ Verificar creación automática de cuenta  
✓ Verificar datos de la tarjeta  
✓ Verificar información de la cuenta  
✓ Visualizar Préstamos Activos

https://frontend-react-homebanking.onrender.com

Command	Target	Value
4 click	id:row	
5 click	id:source-account	
6 select	id:source-account	label:VIN24117271
7 click	id:destinacion-account	
8 select	id:destinacion-account	label:VIN41973894
9 click	id:amount-transaction	
10 type	id:amount-transaction	100
11 click	id:description-transactions	
12 type	id:description-transactions	pago
13 click	css:=bg-[#F3C2D32]	
14 click	css:=px-4 nth-child(1)	

Command: open  #

Target: /

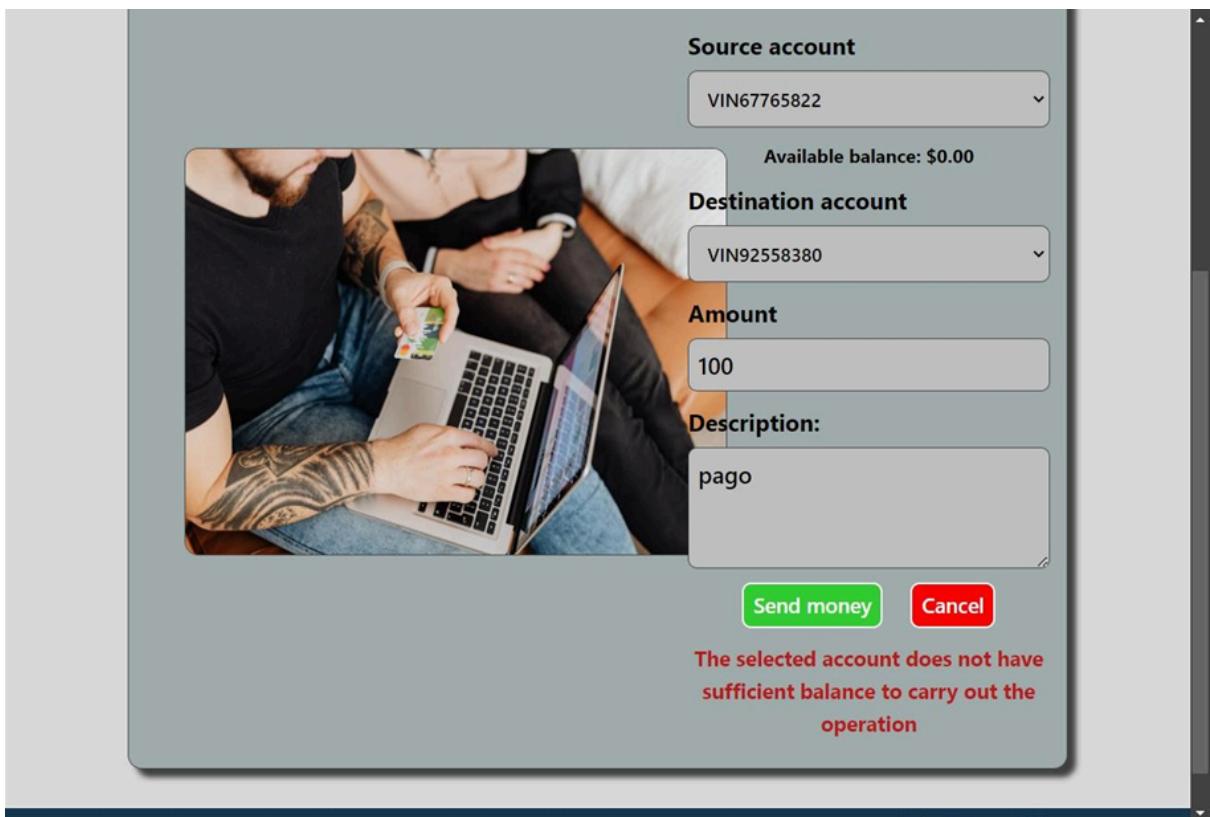
Value:

Description:

Log Reference

13. click on css=border-[#B0B0B0] OK  
14. click on css=bg-[#F3C2D32] OK  
'Registrar un nuevo cliente' completed successfully

21:15:45  
21:15:48  
21:15:49



## Datos de tarjeta:

Selenium IDE - CHALLENGE FINAL \*

Project: CHALLENGE FINAL \*

Tests + https://frontend-react-homebanking.onrender.com

Command	Target	Value
1 open	/	
2 set window size	1070x824	
3 mouse over	linkText:Accounts	
4 mouse out	linkText:Accounts	
5 click	linkText:Cards	

Command: open Target: / Value: Description:

Log Reference Running "Validar la creación de una tarjeta de crédito"

1. open on / OK  
2. setWindowSize on 1552x840 OK  
3. click on linkText:Cards OK  
4. runScript on window.scrollTo(0,500) OK  
5. click on css=w-[200px] OK  
6. mouseDown on id=cardType OK  
7. mouseUp on css=min-h-[calc(100vh-23vh)] OK  
8. click on css=min-h-[calc(100vh-23vh)] OK  
9. select on id=cardType with value label=CREDIT OK  
10. click on id=cardMembership OK

20.06.59  
20.07.00  
20.07.02  
20.07.04  
20.07.07  
20.07.10  
20.07.13  
20.07.16  
20.07.19  
20.07.22  
20.07.25



**Welcome to your card space!**

Here you can manage all your cards easily and quickly. Check the details of your current cards and take the opportunity to request a new one. Your finances at a click!



**Credit**

**Debit**



**Credit Card**

**4839-7931-6959-5900**



**Debit Card**

**0211-2918-4728-6525**

## **Lista de cuenta:**

Selenium IDE - CHALLENGE FINAL \*

Project: CHALLENGE FINAL \*

Tests +

Search tests...

✓ Registrar un nuevo cliente  
✓ Validar el proceso de llenado del formulario mis propias  
✓ Validar funcionalidad de cierre de sesión exitoso  
✓ Validar inicio de sesión  
✓ Validar la creación de una tarjeta de crédito  
✓ Validar la creación de una tarjeta de débito  
✓ Verificar creación automática de cuenta  
✓ Verificar datos de la tarjeta  
✓ Verificar información de la cuenta  
✓ Visualizar Préstamos Activos

https://frontend-react-homebanking.onrender.com

Command	Target	Value
1 open	/	
2 set window size	1070x624	
3 click	linkText:Accounts	

Command: open

Target: /

Value:

Description:

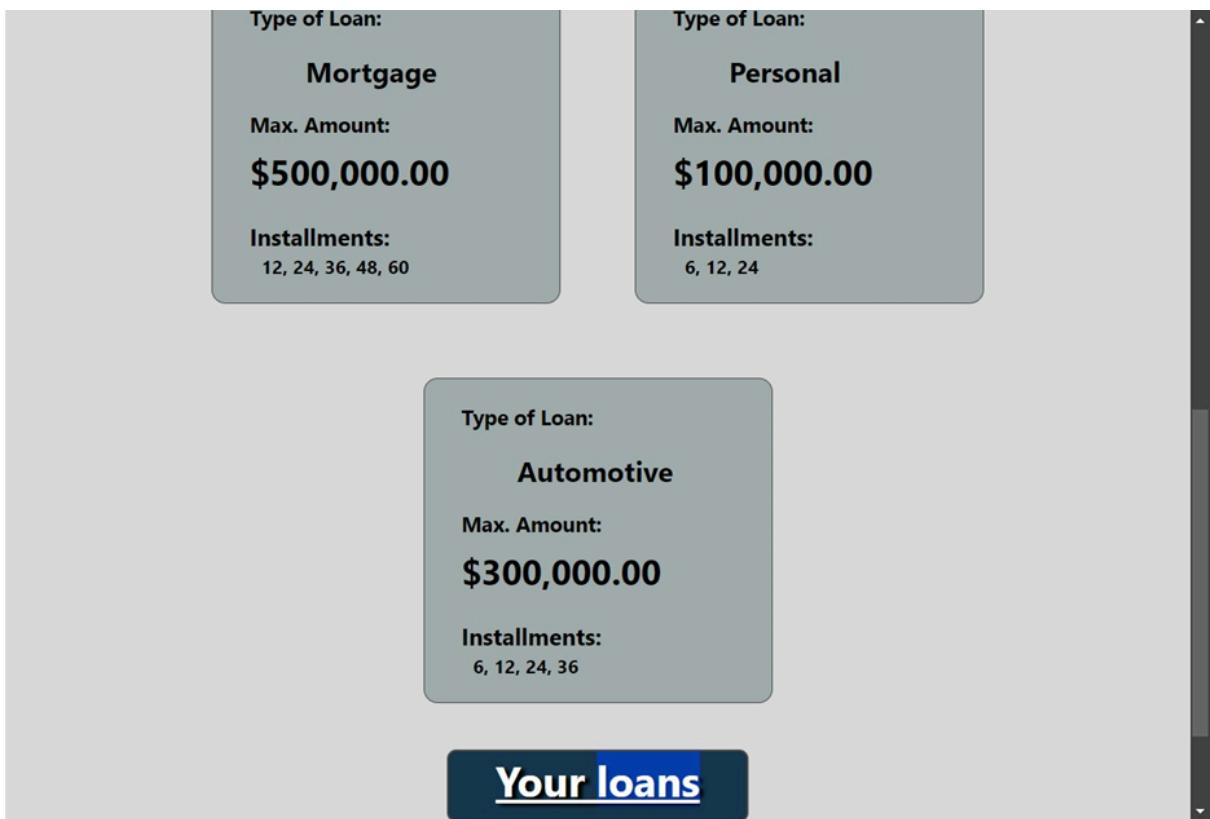
Log Reference

Running 'Validar la creación de una tarjeta de crédito'

Step	Message	Time
1	open on /OK	20:06:59
2	setWindowSize on 1552x640 OK	20:07:00
3	click on linkText:Cards OK	20:07:02
4	runScript on window scrollTo(0,500) OK	20:07:04
5	click on css= w-[200px] OK	20:07:07
6	mouseDown on id:cardType OK	20:07:13
7	mouseUp on css= min-h-[calc((100vh-23vh))] OK	20:07:16
8	click on css= min-h-[calc((100vh-23vh))] OK	20:07:19
9	select on id:cardType with value label=CREDIT OK	20:07:22
10	click on id:cardMembership OK	20:07:25

The image shows the homepage of the Bank Ease website. At the top left is the logo 'BANK EASE BANK SMART' with a stylized building icon. To the right are four yellow buttons labeled 'Accounts', 'Cards', 'Loans', and 'Transactions'. Further right is a white button with a black arrow pointing right, and the text 'Log out' below it. The main title 'Welcome Melba Liliana Lorenzo!' is displayed in a large, bold, white font inside a dark blue rounded rectangle. Below the title is a photograph of a person's hands working at a desk with papers and a calculator. At the bottom left, the text 'Welcome to Bank Ease!' is underlined, followed by the instruction 'On this page you can:'. On the bottom right is a small inset image showing a hand putting a coin into a piggy bank.

## Préstamos no existentes:



## Préstamos activos:

The image shows the Selenium IDE interface with a recorded test case. The test case consists of the following steps:

- open /
- set window size: 1552x840
- click on linkText=Loans

The "Log" tab at the bottom shows the execution details:

```
Running "Validar la creación de una tarjeta de crédito"
1. open on /OK
2. setWindowSize on 1552x840 OK
3. click on linkText=Cards OK
4. runScript on window.scrollTo(0,500) OK
5. click on css=w-[200px] OK
6. mouseDown on id=cardType OK
7. mouseUp on css=min-h-[calc(100vh-23vh)] OK
8. click on css=min-h-[calc(100vh-23vh)] OK
9. select on id=cardType with value label=CREDIT OK
10. click on id=cardMembership OK
```

## Cierre de sesión:

Selenium IDE - CHALLENGE FINAL \*

Project: CHALLENGE FINAL \* [Edit](#)

Tests [New](#) [Import](#) [Export](#) [Delete](#) [Help](#)

Search tests:  [Search](#)

https://frontend-react-homebanking.onrender.com

Command	Target	Value
1 open	/	
2 set window size	1070x824	css:w-1070px-h-824px
3 click	css=w-[\${vw}]	
4 close		

Command:

Target:  [Search](#)

Value:

Description:

Log Reference

Running: Validar la creación de una tarjeta de crédito

1. open on / OK  
2. setWindowSize on 1552x840 OK  
3. click on linkText=Cards OK  
4. runScript on window scrollTo(0,500) OK  
5. click on css=w-[200px] OK  
6. mouseDown on id=cardType OK  
7. mouseUp on css=min-height-[calc(100vh-23vh)] OK  
8. click on css=min-height-[calc(100vh-23vh)] OK  
9. select on id=cardType with value label=CREDIT OK  
10. click on id=cardMembership OK

20.06.59  
20.07.00  
20.07.02  
20.07.04  
20.07.07  
20.07.10  
20.07.13  
20.07.16  
20.07.19  
20.07.22  
20.07.25

The image is a composite of two screenshots. On the left, there is a screenshot of a web-based login form for 'BANK EASE BANK SMART'. The logo features a stylized building with a yellow roof and a dollar sign inside. The form includes fields for 'Email' (with placeholder 'Ex. client@example.com') and 'Password' (with placeholder '\*\*\*\*\*'). Below these are buttons for 'Sign in' and 'Register'. At the bottom, a footer states '© 2024 Bank Ease. All rights reserved.' and 'Cohort 55 -William Ocanto'. On the right, there is a photograph of a person's hands interacting with a silver laptop keyboard. A gold credit card is held in one hand, and a smartphone and a pair of white earphones are visible in the background.

## Análisis de UX/UI

### DISEÑO SITIO WEB

#### Análisis de Accesibilidad:

##### Sugerencias Generales basáandonos en el reporte de lighthouse

- **Caching:** Implementar estrategias de caché para reducir la carga de recursos repetidos en visitas posteriores.
- **Optimización de Imágenes:** Comprimir y ajustar el formato de las imágenes (por ejemplo, usar formatos modernos como WebP-> **JPG y PNG**)
- **Accesibilidad y cumplimiento de estándares:** Se propone agregar el botón para visualizar contraseña este con el fin de evitar intentos de contraseña no válidos.

**Inclusión:** Notamos que la página no es inclusiva con las personas no videntes, en esta nueva versión sugerimos agregar un Box narrativo que ofrezca una descripción de la página, sus herramientas y sus acciones. (Versión móvil)

**Diseño y vista:** Los colores del home banking no son muy profesionales ni agradables a la vista, su diseño es poco profesional brindando una apariencia “Cruda”, se propone crear una nueva versión rediseñada, con una temática más llamativa, también agregar la opción de Light Mode o Dark mode.

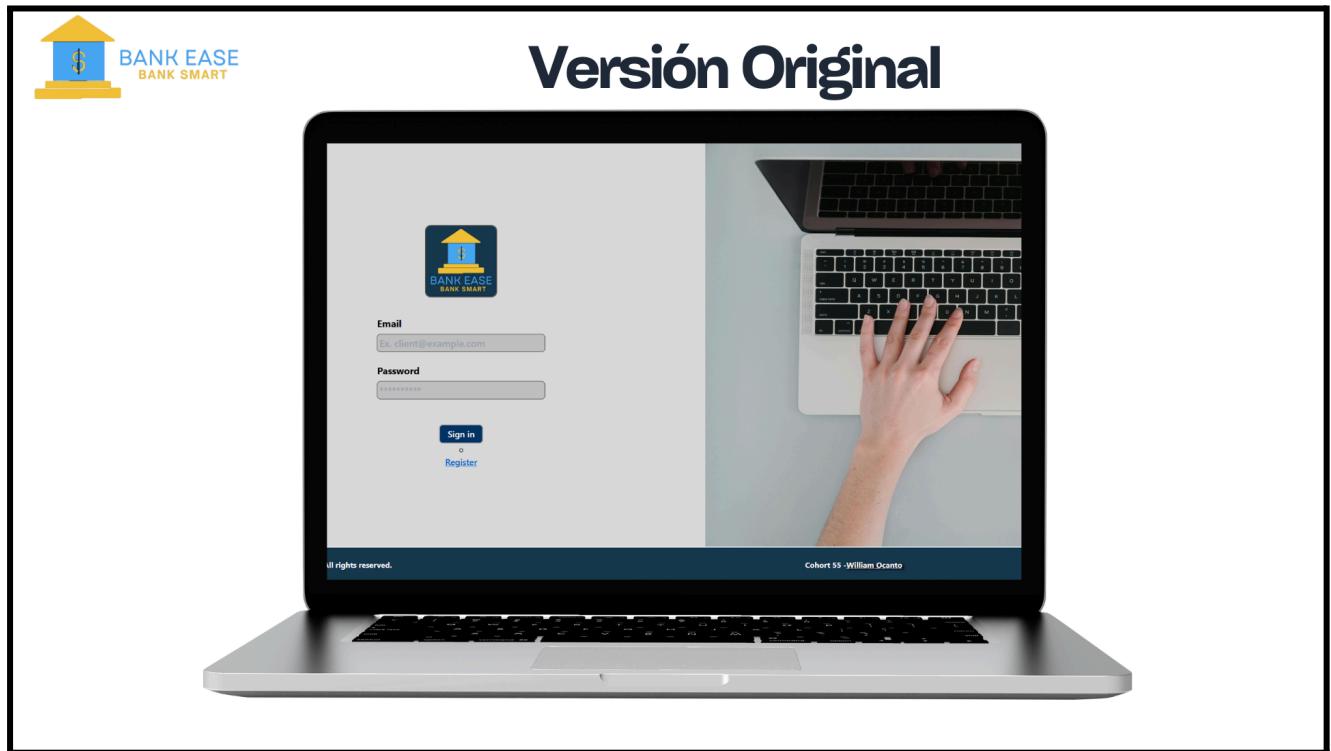
- **Diseño:** El diseño actual no tiene un contraste de color adecuado, textos alternativos para imágenes, o navegación accesible con teclado, es posible que se esté excluyendo a una parte significativa de la audiencia. Se recomienda mejorar estos aspectos y así ampliar el alcance de la página y demostrar un compromiso con la inclusión.
- **PQRS, FAQ y sistema de soporte:** La página no cuenta con espacios de ayuda para el usuario, lo cual hace que su experiencia no tenga el acompañamiento adecuado en el momento en que se presente una inconformidad, tampoco permite radicar solicitudes como documentación, fallos o incongruencias de funcionamiento. No tiene disponible un segmento de calificación de experiencia que permita realizar un sondeo de satisfacción. Tampoco tiene líneas de atención al cliente para llamadas de soporte ni chat en vivo. Por ende se sugiere en la nueva versión propuesta agregar este segmento
- **Recomendaciones para la implementación de seguridad:** Se propone implementar un sistema de recuperación de cuenta, ya que este es importante para brindar confiabilidad a los usuarios: se propone implementar

**Envío de enlaces de recuperación:** Utilizar un sistema que envíe un enlace seguro al correo electrónico registrado del usuario para restablecer su contraseña.

**Verificación por SMS:** Ofrecer la opción de recuperación a través de un código de verificación enviado por SMS como una capa adicional de seguridad.

**Preguntas de seguridad opcionales:** Añadir preguntas de seguridad configurables por el usuario para fortalecer el proceso de recuperación.

**Cierre de sesión automático (auto-logout):** añadir un sistema de auto-logout de 5 min a 15 min es esencial para la seguridad de la web.



## REDISEÑO

### Propuesta de Diseño web

The proposed web design for the login page features a modern aesthetic with a dark blue background. On the left, there's a decorative graphic element resembling a stylized globe or a brain. The right side contains the 'BANK EASE' logo in large, bold, purple letters. Below the logo is the login form with 'Email' and 'Password' fields, and 'Sing In', 'Reset Account', and 'Register' buttons. The footer includes the text '2024 Bank Ease. All rights reserved.'

### Propuesta de diseño móvil

The proposed mobile design for the login page is a simplified version of the web design. It features a dark blue background with the 'Bank Ease' logo at the top. The login form with 'Email' and 'Password' fields, and 'Sing In', 'Reset Account', and 'Register' buttons is centered. The footer includes the text '© 2024 Bank Ease. All rights reserved.'

A small version of the 'BANK EASE BANK SMART' logo, featuring the stylized building icon and the text 'BANK EASE BANK SMART'.

## Conclusiones

Durante las pruebas llevadas a cabo en la web del home banking, se han identificado ciertos aspectos que requieren ajustes antes de considerar el despliegue a la producción. El sistema aún presenta inconsistencias en áreas críticas de usabilidad, accesibilidad y desempeño.

Concluimos que los principales hallazgos son: tiempos de carga elevados, inconvenientes en el flujo de transacciones y en la visualización de la información del cliente, lo que afecta la precisión, fiabilidad del sistema y la experiencia de usuario. Es esencial realizar nuevas rondas de testing después de implementar las mejoras sugeridas, con el fin de que el sistema cumpla con los estándares de calidad establecidos y requeridos por la industria bancaria.

## Links

- Proyecto Jira:  
<https://lauracastrog.atlassian.net/jira/software/projects/CHAL/boards/1/backlog>
- Confluence:  
<https://lauracastrog.atlassian.net/wiki/external/Y2ZjZDM1MDExMWEwNGZhNDliODYwZTA5YTFkOTEzYjk>
- Presentación:  
<https://bit.ly/QAequipo4>

