

Informe de Pruebas Técnicas

Proyecto: Sistema de Carga de Nómina Bancaria

Desarrollador: Victor Tama

Fecha: 13 de Enero de 2026

Tecnologías: Java 21 (Spring Boot) / Angular 19 / H2 Database

1. Objetivo

El presente documento certifica la correcta implementación y funcionamiento del sistema de carga masiva de archivos planos para la creación de clientes y cuentas bancarias, validando la integridad de los datos, la integración con servicios externos (Databook simulado) y la persistencia en base de datos.

2. Alcance de las Pruebas

Se han ejecutado pruebas unitarias e integrales cubriendo los siguientes escenarios:

- Conexión y disponibilidad de la API (Backend).
- Interfaz de usuario y carga de archivos (Frontend).
- Validaciones de negocio (Reglas de validación).
- Integración con Base de Datos H2.

3. Casos de Prueba Ejecutados


Caso 1: Carga de Archivo con Datos Válidos

Descripción: Se sube un archivo .txt que contiene un registro que cumple con todas las validaciones (formato correcto, email válido, celular de 10 dígitos).

- Entrada:** Registro C|1234567890|2023-10-25|500.50|valido@banco.com|0987654321
- Resultado Esperado:** El sistema procesa la línea, consulta el Databook para obtener nombres, y guarda el registro en la base de datos. El Frontend muestra "Exitosos: 1".

Evidencia (Frontend):


localhost:4200

 Sistema de Carga de Nómina


Seleccione el archivo plano (TXT):

Elegir archivo

No se eligió ningún archivo

 Subir Archivo


localhost:4200

 Sistema de Carga de Nómina


Seleccione el archivo plano (TXT):


Elegir archivo

prueba_clientes.txt

 Subir Archivo

 Reporte de Carga

 Exitosos: 1

 Errores: 2

Detalle de errores:

- Línea 2: El número de celular valido debe ser de 10 dígitos numéricos.
- Línea 3: El formato del correo electrónico no es correcto, reviselo.

Evidencia (Base de Datos):

← ↻ ⓘ localhost:8080/h2-console/login.do?jsessionid=824d8210aed39a2f7c560f1bfb160e52

🔊 🔍 ☒ Auto commit 🔍 ☐ Max rows: 1000 Auto complete Off

📁 jdbc:h2:mem:nominadb

📁 CLIENTES

⊕ 📄 ID

⊕ 📄 APELLIDOS

⊕ 📄 CORREO_ELECTRONICO

⊕ 📄 FECHA_INGRESO

⊕ 📄 FECHA_NACIMIENTO

⊕ 📄 NOMBRES

⊕ 📄 NUMERO_CELULAR

⊕ 📄 NUMERO_CUENTA

⊕ 📄 NUMERO_IDENTIFICACION

⊕ 📄 TIPO_IDENTIFICACION

⊕ 📄 VALOR_NOMINA

⊕ 📄 Indexes

📁 INFORMATION_SCHEMA

👤 Users

🕒 H2 2.4.240 (2025-09-22)

SQL statement:

Important Commands

?		Displays this Help Page
📄		Shows the Command History
🏃	Ctrl+Enter	Executes the current SQL statement
🏃	Shift+Enter	Executes the SQL statement defined by the text selection
	Ctrl+Space	Auto complete
🔊		Disconnects from the database

← ↻ ⓘ localhost:8080/h2-console/login.do?jsessionid=824d8210aed39a2f7c560f1bfb160e52

🔊 🔍 ☒ Auto commit 🔍 ☐ Max rows: 1000 Auto complete Off Auto select On

📁 jdbc:h2:mem:nominadb

📁 CLIENTES

⊕ 📄 ID

⊕ 📄 APELLIDOS

⊕ 📄 CORREO_ELECTRONICO

⊕ 📄 FECHA_INGRESO

⊕ 📄 FECHA_NACIMIENTO

⊕ 📄 NOMBRES

⊕ 📄 NUMERO_CELULAR

⊕ 📄 NUMERO_CUENTA

⊕ 📄 NUMERO_IDENTIFICACION

⊕ 📄 TIPO_IDENTIFICACION

⊕ 📄 VALOR_NOMINA

⊕ 📄 Indexes

📁 INFORMATION_SCHEMA

👤 Users

🕒 H2 2.4.240 (2025-09-22)

SQL statement:

select * from clientes

select * from clientes;

ID	APELLIDOS	CORREO_ELECTRONICO	FECHA_INGRESO	FECHA_NACIMIENTO	NOMBRES	NUMERO_CELULAR	NUMERO_CUENTA	NUMERO_IDENTIFICACION	TIPO_IDENTIFICACION
1	Tama Perez	valido@banco.com	2023-10-25	1983-17-15	Victor Javier	0987654321	CTA-1768340718663-7890	1234567890	C

(1 row, 3 ms)

Caso 2: Validación de Errores de Formato

Descripción: Se valida que el sistema rechace registros que no cumplan con las reglas de negocio bancarias.


- **Entrada:**
 1. Registro con celular incorrecto (longitud diferente a 10).
 2. Registro con email inválido (sin @ o dominio).
- **Resultado Esperado:** El sistema no guarda estos registros y devuelve una lista detallada de errores.

Evidencia (Reporte de Errores en UI):

Sistema de Carga de Nómina

Seleccione el archivo plano (TXT):

Prueba Téc...ntes - Em.txt

 Subir Archivo

Reporte de Carga

✓ Exitosos: 0

✗ Errores: 52

Detalle de errores:

- Línea 1: Formato incompleto. (Columnas: 1)
- Línea 2: Formato incompleto. (Columnas: 1)
- Línea 3: Formato incompleto. (Columnas: 1)
- Línea 5: Formato incompleto. (Columnas: 1)
- Línea 7: Formato incompleto. (Columnas: 1)
- Línea 8: Formato incompleto. (Columnas: 1)
- Línea 10: Formato incompleto. (Columnas: 1)
- Línea 12: Formato incompleto. (Columnas: 1)

Caso 3: Validación de Estructura de Archivo

Descripción: El sistema debe detectar líneas incompletas o vacías en el archivo de texto.

- **Entrada:** Una línea con menos columnas de las requeridas (ej: falta el celular).
- **Resultado Esperado:** El backend rechaza la línea específica indicando "Formato incompleto" sin detener la carga de los demás registros.

Evidencia:

Code	Details
200	<div>Response body</div> <pre>{ "exitosos": 0, "errores": ["Línea 1: Formato incompleto. (Columnas: 1)", "Línea 2: Formato incompleto. (Columnas: 1)", "Línea 3: Formato incompleto. (Columnas: 1)", "Línea 5: Formato incompleto. (Columnas: 1)", "Línea 7: Formato incompleto. (Columnas: 1)", "Línea 8: Formato incompleto. (Columnas: 1)", "Línea 10: Formato incompleto. (Columnas: 1)", "Línea 12: Formato incompleto. (Columnas: 1)", "Línea 14: Formato incompleto. (Columnas: 1)", "Línea 16: Formato incompleto. (Columnas: 1)", "Línea 18: Formato incompleto. (Columnas: 1)", "Línea 20: Formato incompleto. (Columnas: 1)", "Línea 22: Formato incompleto. (Columnas: 1)", "Línea 23: Formato incompleto. (Columnas: 1)", "Línea 25: Formato incompleto. (Columnas: 1)", "Línea 27: Formato incompleto. (Columnas: 1)", "Línea 29: Formato incompleto. (Columnas: 1)", "Línea 31: Formato incompleto. (Columnas: 1)", "Línea 33: Formato incompleto. (Columnas: 1)", "Línea 35: Formato incompleto. (Columnas: 1)", "Línea 37: Formato incompleto. (Columnas: 1)",] }</pre>

Caso 4: Documentación de API

Descripción: Verificación de que los endpoints están correctamente documentados y accesibles mediante OpenAPI.

Evidencia:

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/clientes/cargar-archivo' \
  -H 'accept: */*' \
  -H 'Content-Type: multipart/form-data' \
  -F 'archivo=@prueba_clientes.txt;type-text/plain'
```

Request URL

http://localhost:8080/api/clientes/cargar-archivo

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "exitosos": 0, "errores": ["Línea 1: El cliente con identificación 1234567890 ya existe en la base de datos.", "Línea 2: El número de celular válido debe ser de 10 dígitos numéricos.", "Línea 3: El formato del correo electrónico no es correcto, revíselo."] }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>connection: keep-alive content-type: application/json date: Tue, 13 Jun 2023 21:51:26 GMT keep-alive: timeout=60 transfer-encoding: chunked vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers</pre></div></div>

Responses

Code	Description	Links
------	-------------	-------