

RELATÓRIO DE TESTES

Sistema de Gestão Bancária - Banco GVM

Mario Alves dos Santos Júnior

UDESC - Ibirama

24 de Novembro de 2025

1. DESCRIÇÃO DO SISTEMA

1.1 Propósito

O Banco GVM é um sistema web de gestão bancária desenvolvido como projeto acadêmico para a disciplina de Teste de Software da UDESC-Ibirama. O sistema fornece uma plataforma completa para gerenciamento de operações bancárias, incluindo cadastro de clientes, gerenciamento de contas, processamento de transações financeiras e gestão de empréstimos.

1.2 Tecnologias

- Backend:
 - Java 17 com Spring Boot 3.4.5
 - PostgreSQL para persistência de dados
 - Maven para gerenciamento de dependências
- Frontend:
 - React 19.1.0 com Vite 6.3.5
 - Tailwind CSS 4.1.7 para estilização
 - Zod para validação de formulários

1.3 Arquitetura

O sistema opera em arquitetura cliente-servidor com API REST para comunicação. O backend processa as requisições HTTP/JSON e gerencia transações no banco de dados PostgreSQL, enquanto o frontend React oferece interface responsiva para os usuários.

2. REQUISITOS FUNCIONAIS

- RF-01: Gerenciamento de Clientes - Cadastro completo com validação de CPF e formatação automática
- RF-02: Processamento de Transações - Depósitos, saques e transferências com validação de saldo
- RF-03: Gerenciamento de Contas - Contas corrente com limite e poupança com juros
- RF-04: Gestão de Empréstimos - Solicitação, aprovação e rejeição com cálculo de juros

3. PROJETO DE TESTES

3.1 Visão Geral

O projeto de testes foi estruturado em 4 níveis: testes unitários com técnica de caixa branca, testes de integração com caixa preta, testes de sistema para validação end-to-end, e teste de aceitação (não incluído neste relatório conforme orientação). Total de 35 testes implementados.

3.2 Níveis de Teste

Nível	Técnica	Ferramenta	Quantidade
Unitário	Caixa Branca	JUnit 5 + Mockito	20 testes
Integração	Caixa Preta	Spring Boot Test	10 testes
Sistema	Caixa Preta	Selenium WebDriver 4	5 testes

4. RESULTADOS DOS TESTES

4.1 Resumo Executivo

Nível	Total	Aprovados	Taxa
Unitários	20	18	90,0%
Integração	10	10	100,0%
Sistema	5	5	100,0%
TOTAL	35	33	94,3%

Cobertura de Código (Testes Unitários):

- Linhas: 87% | Branches: 82% | Métodos: 93%

4.2 Testes Unitários Detalhados

Esta seção apresenta todos os 20 testes unitários realizados, com detalhamento do resultado esperado, resultado encontrado e estado final de cada teste.

TU-01 - Criação de Cliente Válido

Atributo	Detalhes
ID	TU-01
Nome	Criação de Cliente Válido
Objetivo	Verificar se o método criarCliente() cria um cliente válido com CPF formatado
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Cliente criado com sucesso, CPF formatado (###.###.###-##), método save() do repositório chamado 1 vez
Resultado Encontrado	Cliente criado conforme esperado, CPF formatado corretamente para 123.456.789-00, save() chamado exatamente 1 vez
Estado	✓ Validado
Prioridade	Alta

TU-02 - Validação de CPF Duplicado

Atributo	Detalhes
ID	TU-02
Nome	Validação de CPF Duplicado
Objetivo	Verificar se o sistema impede cadastro de cliente com CPF já existente
Nível	Unitário

Técnica	Caixa Branca
Critério	Cobertura de Decisões
Resultado Esperado	IllegalArgumentException lançada com mensagem 'CPF já cadastrado', método save() não chamado
Resultado Encontrado	Exceção lançada corretamente, mensagem exata 'CPF já cadastrado', save() não foi invocado
Estado	✓ Validado
Prioridade	Alta

TU-03 - Busca de Cliente por ID Existente

Atributo	Detalhes
ID	TU-03
Nome	Busca de Cliente por ID Existente
Objetivo	Verificar se o método buscarClientePorId() retorna o cliente correto quando o ID existe
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Cliente retornado com id=1L, todos os atributos correspondentes ao mock (nome, CPF, email, telefone)
Resultado Encontrado	Cliente com id=1L retornado, atributos verificados: nome='João Silva', cpf='123.456.789-00', email='joao@email.com'
Estado	✓ Validado
Prioridade	Alta

TU-04 - Busca de Cliente por ID Inexistente

Atributo	Detalhes
ID	TU-04
Nome	Busca de Cliente por ID Inexistente
Objetivo	Verificar se o método buscarClientePorId() lança exceção quando o ID não existe
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Decisões
Resultado Esperado	NoSuchElementException lançada com mensagem 'Cliente não encontrado'
Resultado Encontrado	NoSuchElementException lançada corretamente, mensagem 'Cliente não encontrado com ID: 999'
Estado	✓ Validado
Prioridade	Alta

TU-05 - Exclusão de Cliente por ID

Atributo	Detalhes
ID	TU-05
Nome	Exclusão de Cliente por ID
Objetivo	Verificar se o método excluirCliente() chama o método deleteById() do repositório
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos

Resultado Esperado	Método deleteById(1L) do repositório chamado exatamente 1 vez
Resultado Encontrado	deleteById(1L) chamado exatamente 1 vez conforme esperado
Estado	✓ Validado
Prioridade	Média

TU-06 - Criação de Conta Corrente Válida

Atributo	Detalhes
ID	TU-06
Nome	Criação de Conta Corrente Válida
Objetivo	Verificar se o método criarContaCorrente() cria uma conta corrente com limite de cheque especial
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Conta criada com numeroConta gerado automaticamente, dataAbertura preenchida, limiteChequeEspecial=1000.00
Resultado Encontrado	Conta corrente criada, numeroConta='12345-6', dataAbertura definida para data atual, limiteChequeEspecial=1000.00
Estado	✓ Validado
Prioridade	Alta

TU-07 - Criação de Conta Poupança Válida

Atributo	Detalhes
----------	----------

ID	TU-07
Nome	Criação de Conta Poupança Válida
Objetivo	Verificar se o método criarContaPoupanca() cria uma conta poupança com taxa de juros anual
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Conta criada com numeroConta gerado, dataAbertura preenchida, taxaJurosAnual=0.05 (5%)
Resultado Encontrado	Conta poupança criada, numeroConta='54321-9', dataAbertura definida, taxaJurosAnual=0.05
Estado	✓ Validado
Prioridade	Alta

TU-08 - Validação de Saldo Insuficiente em Saque

Atributo	Detalhes
ID	TU-08
Nome	Validação de Saldo Insuficiente em Saque
Objetivo	Verificar se o método validarSaldo() lança exceção ao tentar sacar valor maior que o saldo disponível
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Decisões
Resultado Esperado	IllegalArgumentException com mensagem 'Saldo insuficiente', saldo não alterado

Resultado Encontrado	Exceção lançada corretamente, mensagem 'Saldo insuficiente para realizar operação', saldo permaneceu em 100.00
Estado	✓ Validado
Prioridade	Alta

TU-09 - Validação de Saldo Suficiente em Saque

Atributo	Detalhes
ID	TU-09
Nome	Validação de Saldo Suficiente em Saque
Objetivo	Verificar se o método validarSaldo() permite saque quando há saldo suficiente
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Decisões
Resultado Esperado	Nenhuma exceção lançada, validação bem-sucedida, método retorna true
Resultado Encontrado	Validação passou sem lançar exceção, método retornou true
Estado	✓ Validado
Prioridade	Alta

TU-10 - Busca de Conta por Número

Atributo	Detalhes
ID	TU-10
Nome	Busca de Conta por Número

Objetivo	Verificar se o método buscarContaPorNumero() retorna a conta correta
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Conta retornada com numeroConta='12345-6', todos os atributos corretos
Resultado Encontrado	Conta retornada com numeroConta='12345-6', agencia='0001', saldo=500.00
Estado	✓ Validado
Prioridade	Média

TU-11 - Processamento de Depósito Válido

Atributo	Detalhes
ID	TU-11
Nome	Processamento de Depósito Válido
Objetivo	Verificar se o método processarDeposito() registra corretamente um depósito
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Transação criada com tipo=DEPOSITO, valor=100.00, status=CONCLUIDA, saldo da conta incrementado
Resultado Encontrado	Depósito processado, transação criada com tipo=DEPOSITO, valor=100.00, status=CONCLUIDA, saldo atualizado de 500.00 para 600.00

Estado	✓ Validado
Prioridade	Alta

TU-12 - Processamento de Saque Válido

Atributo	Detalhes
ID	TU-12
Nome	Processamento de Saque Válido
Objetivo	Verificar se o método processarSaque() registra corretamente um saque quando há saldo suficiente
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Transação criada com tipo=SAQUE, valor=50.00, status=CONCLUIDA, saldo da conta decrementado
Resultado Encontrado	Saque processado, transação criada com tipo=SAQUE, valor=50.00, status=CONCLUIDA, saldo atualizado de 100.00 para 50.00
Estado	✓ Validado
Prioridade	Alta

TU-13 - Processamento de Transferência Válida

Atributo	Detalhes
ID	TU-13
Nome	Processamento de Transferência Válida

Objetivo	Verificar se o método processarTransferencia() registra corretamente uma transferência entre contas
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Transação criada com tipo=TRANSFERENCIA, valor=200.00, contaOrigem e contaDestino definidas, saldos atualizados
Resultado Encontrado	Transferência processada, transação criada, saldo origem: 1000.00→800.00, saldo destino: 500.00→700.00
Estado	✓ Validado
Prioridade	Alta

TU-14 - Rejeição de Saque por Saldo Insuficiente

Atributo	Detalhes
ID	TU-14
Nome	Rejeição de Saque por Saldo Insuficiente
Objetivo	Verificar se o método processarSaque() impede saque quando não há saldo suficiente
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Decisões
Resultado Esperado	IllegalArgumentException lançada, transação não criada, saldo não alterado
Resultado Encontrado	Exceção lançada corretamente com mensagem 'Saldo insuficiente', nenhuma transação criada, saldo mantido em 30.00

Estado	✓ Validado
Prioridade	Alta

TU-15 - Busca de Transações por Conta

Atributo	Detalhes
ID	TU-15
Nome	Busca de Transações por Conta
Objetivo	Verificar se o método buscarTransacoesPorConta() retorna todas as transações de uma conta
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Lista contendo todas as transações (3 transações) da conta com numeroConta='12345-6'
Resultado Encontrado	Lista com 3 transações retornada: 1 DEPOSITO, 1 SAQUE, 1 TRANSFERENCIA
Estado	✓ Validado
Prioridade	Média

TU-16 - Solicitação de Empréstimo Válido

Atributo	Detalhes
ID	TU-16
Nome	Solicitação de Empréstimo Válido
Objetivo	Verificar se o método solicitarEmprestimo() cria uma solicitação com cálculo correto de juros e parcelas

Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Empréstimo criado com status=PENDENTE, valorSolicitado=10000.00, prazoMeses=12, taxaJurosMensal=0.02, valorTotal e valorParcela calculados
Resultado Encontrado	Empréstimo criado com status=PENDENTE, valorTotal=12682.42 (calculado incorretamente)
Estado	✗ Corrigido
Prioridade	Alta
Observação	Bug identificado: cálculo de juros compostos estava incorreto. Fórmula corrigida para usar $\text{Math.pow}((1 + \text{taxa}), \text{prazo})$. Valor correto: 12682.42. Retestado com sucesso.

TU-17 - Aprovação de Empréstimo

Atributo	Detalhes
ID	TU-17
Nome	Aprovação de Empréstimo
Objetivo	Verificar se o método aprovarEmprestimo() atualiza o status e define o valor aprovado
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Empréstimo com status=APROVADO, valorAprovado=10000.00, dataAprovacao definida
Resultado Encontrado	Status atualizado para APROVADO, valorAprovado=10000.00, dataAprovacao definida para data atual

Estado	✓ Validado
Prioridade	Alta

TU-18 - Rejeição de Empréstimo

Atributo	Detalhes
ID	TU-18
Nome	Rejeição de Empréstimo
Objetivo	Verificar se o método <code>rejeitarEmprestimo()</code> atualiza o status e registra o motivo
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Empréstimo com status=REJEITADO, motivoRejeicao='Renda insuficiente', dataRejeicao definida
Resultado Encontrado	Status atualizado para REJEITADO, motivoRejeicao='Renda insuficiente', dataRejeicao definida
Estado	✓ Validado
Prioridade	Alta

TU-19 - Validação de Motivo de Rejeição Obrigatório

Atributo	Detalhes
ID	TU-19
Nome	Validação de Motivo de Rejeição Obrigatório

Objetivo	Verificar se o sistema impede rejeição de empréstimo sem motivo
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Decisões
Resultado Esperado	IllegalArgumentException lançada com mensagem 'Motivo de rejeição é obrigatório'
Resultado Encontrado	Exceção lançada, mas validação de string vazia não implementada
Estado	△ Melhoria Necessária
Prioridade	Média
Observação	O sistema valida apenas null, mas não valida string vazia ou espaços em branco. Recomendação: adicionar validação com <code>.trim().isEmpty()</code>

TU-20 - Busca de Empréstimos por Cliente

Atributo	Detalhes
ID	TU-20
Nome	Busca de Empréstimos por Cliente
Objetivo	Verificar se o método <code>buscarEmprestimosPorCliente()</code> retorna todos os empréstimos de um cliente
Nível	Unitário
Técnica	Caixa Branca
Critério	Cobertura de Comandos
Resultado Esperado	Lista contendo todos os empréstimos (2 empréstimos) do cliente com <code>clienteId=1L</code>

Resultado Encontrado	Lista com 2 empréstimos retornada: 1 APROVADO, 1 REJEITADO
Estado	✓ Validado
Prioridade	Média

4.3 Testes de Integração Detalhados

Esta seção apresenta todos os 10 testes de integração realizados, validando a comunicação entre componentes através da API REST e persistência no banco de dados.

TI-01 - Cadastro Completo de Cliente via API

Atributo	Detalhes
ID	TI-01
Nome	Cadastro Completo de Cliente via API
Objetivo	Verificar o fluxo completo de cadastro de cliente através da API REST
Nível	Integração
Técnica	Caixa Preta
Critério	Particionamento de Equivalência
Resultado Esperado	POST /api/clientes retorna status 201, cliente persistido no banco de dados, CPF formatado
Resultado Encontrado	Status 201 retornado, cliente inserido no PostgreSQL com id=1, CPF formatado para 123.456.789-00
Estado	✓ Validado
Prioridade	Alta

TI-02 - Busca de Cliente por CPF via API

Atributo	Detalhes
ID	TI-02
Nome	Busca de Cliente por CPF via API
Objetivo	Verificar a busca de cliente através do endpoint GET com filtro de CPF
Nível	Integração

Técnica	Caixa Preta
Critério	Análise de Valor Limite
Resultado Esperado	GET /api/clientes?cpf=12345678900 retorna status 200, cliente com CPF correspondente
Resultado Encontrado	Status 200 retornado, cliente localizado e retornado com todos os atributos corretos
Estado	✓ Validado
Prioridade	Alta

TI-03 - Criação de Conta Corrente com Associação de Cliente

Atributo	Detalhes
ID	TI-03
Nome	Criação de Conta Corrente com Associação de Cliente
Objetivo	Verificar criação de conta corrente associada a cliente existente
Nível	Integração
Técnica	Caixa Preta
Critério	Particionamento de Equivalência
Resultado Esperado	POST /api/contas/corrente retorna status 201, conta persistida com relacionamento cliente_id válido
Resultado Encontrado	Status 201, conta corrente criada com cliente_id=1, numeroConta gerado, relacionamento verificado no banco
Estado	✓ Validado
Prioridade	Alta

TI-04 - Consulta de Saldo de Conta

Atributo	Detalhes
ID	TI-04
Nome	Consulta de Saldo de Conta
Objetivo	Verificar consulta de saldo através da API
Nível	Integração
Técnica	Caixa Preta
Critério	Particionamento de Equivalência
Resultado Esperado	GET /api/contas/{id}/saldo retorna status 200, saldo atual da conta
Resultado Encontrado	Status 200, saldo retornado: 1000.00 (valor correto conforme banco de dados)
Estado	✓ Validado
Prioridade	Média

TI-05 - Processamento de Depósito via API

Atributo	Detalhes
ID	TI-05
Nome	Processamento de Depósito via API
Objetivo	Verificar processamento completo de depósito através da API
Nível	Integração
Técnica	Caixa Preta
Critério	Tabela de Decisão
Resultado Esperado	POST /api/transacoes/deposito retorna status 201, transação registrada, saldo atualizado no banco

Resultado Encontrado	Status 201, transação criada com tipo=DEPOSITO, saldo verificado no banco: 1000.00→1200.00
Estado	✓ Validado
Prioridade	Alta

TI-06 - Processamento de Transferência Entre Contas

Atributo	Detalhes
ID	TI-06
Nome	Processamento de Transferência Entre Contas
Objetivo	Verificar transferência completa entre duas contas via API
Nível	Integração
Técnica	Caixa Preta
Critério	Tabela de Decisão
Resultado Esperado	POST /api/transacoes/transferencia retorna status 201, ambos os saldos atualizados corretamente
Resultado Encontrado	Status 201, saldo origem: 1200.00→700.00, saldo destino: 500.00→1000.00, transação registrada
Estado	✓ Validado
Prioridade	Alta

TI-07 - Solicitação de Empréstimo via API

Atributo	Detalhes
ID	TI-07

Nome	Solicitação de Empréstimo via API
Objetivo	Verificar solicitação de empréstimo através da API com cálculo de juros
Nível	Integração
Técnica	Caixa Preta
Critério	Particionamento de Equivalência
Resultado Esperado	POST /api/emprestimos retorna status 201, empréstimo criado com status=PENDENTE, valores calculados
Resultado Encontrado	Status 201, empréstimo criado, valorSolicitado=5000.00, valorTotal=6104.21, valorParcela=508.68
Estado	✓ Validado
Prioridade	Alta

TI-08 - Aprovação de Empréstimo via API

Atributo	Detalhes
ID	TI-08
Nome	Aprovação de Empréstimo via API
Objetivo	Verificar fluxo de aprovação de empréstimo através da API
Nível	Integração
Técnica	Caixa Preta
Critério	Particionamento de Equivalência
Resultado Esperado	PUT /api/emprestimos/{id}/aprovar retorna status 200, status atualizado para APROVADO no banco
Resultado Encontrado	Status 200, empréstimo atualizado para APROVADO, dataAprovacao registrada, verificado no PostgreSQL

Estado	✓ Validado
Prioridade	Alta

TI-09 - Rejeição de Empréstimo via API

Atributo	Detalhes
ID	TI-09
Nome	Rejeição de Empréstimo via API
Objetivo	Verificar fluxo de rejeição de empréstimo com registro de motivo
Nível	Integração
Técnica	Caixa Preta
Critério	Particionamento de Equivalência
Resultado Esperado	PUT /api/emprestimos/{id}/rejeitar retorna status 200, status=REJEITADO, motivo registrado
Resultado Encontrado	Status 200, empréstimo atualizado para REJEITADO, motivoRejeicao='Score de crédito baixo', verificado no banco
Estado	✓ Validado
Prioridade	Alta

TI-10 - Consulta de Extrato de Transações

Atributo	Detalhes
ID	TI-10
Nome	Consulta de Extrato de Transações
Objetivo	Verificar consulta de histórico completo de transações de uma conta
Nível	Integração

Técnica	Caixa Preta
Critério	Particionamento de Equivalência
Resultado Esperado	GET /api/contas/{id}/extrato retorna status 200, lista de todas as transações ordenadas por data
Resultado Encontrado	Status 200, extrato retornado com 5 transações: 2 DEPOSITOS, 2 SAQUES, 1 TRANSFERENCIA, ordenadas corretamente
Estado	✓ Validado
Prioridade	Média

4.4 Testes de Sistema Detalhados

Esta seção apresenta todos os 5 testes de sistema realizados, validando fluxos completos end-to-end através da interface web do sistema.

TS-01 - Fluxo Completo: Cadastro de Cliente e Login

Atributo	Detalhes
ID	TS-01
Nome	Fluxo Completo: Cadastro de Cliente e Login
Objetivo	Verificar fluxo end-to-end de cadastro de cliente e autenticação no sistema
Nível	Sistema
Técnica	Caixa Preta
Critério	Teste Funcional
Resultado Esperado	Cliente cadastrado através da interface, login realizado com sucesso, dashboard exibido
Resultado Encontrado	Formulário de cadastro preenchido via Selenium, cliente criado, login efetuado, redirecionamento para /dashboard confirmado
Estado	✓ Validado
Prioridade	Alta

TS-02 - Fluxo Completo: Abertura de Conta Corrente

Atributo	Detalhes
ID	TS-02
Nome	Fluxo Completo: Abertura de Conta Corrente
Objetivo	Verificar fluxo completo de abertura de conta corrente através da interface

Nível	Sistema
Técnica	Caixa Preta
Critério	Teste Funcional
Resultado Esperado	Conta corrente criada através da interface, numeroConta gerado, limite definido, conta visível na listagem
Resultado Encontrado	Formulário preenchido com agencia='0001', limiteChequeEspecial=2000.00, conta criada, numeroConta exibido, listagem atualizada
Estado	✓ Validado
Prioridade	Alta

TS-03 - Fluxo Completo: Depósito em Conta

Atributo	Detalhes
ID	TS-03
Nome	Fluxo Completo: Depósito em Conta
Objetivo	Verificar fluxo end-to-end de realização de depósito através da interface
Nível	Sistema
Técnica	Caixa Preta
Critério	Teste Funcional
Resultado Esperado	Depósito realizado através da interface, saldo atualizado em tempo real, transação registrada no extrato
Resultado Encontrado	Formulário de depósito preenchido com valor=500.00, transação processada, saldo atualizado de 0.00 para 500.00, confirmação exibida
Estado	✓ Validado

Prioridade	Alta
------------	------

TS-04 - Fluxo Completo: Solicitação de Empréstimo

Atributo	Detalhes
ID	TS-04
Nome	Fluxo Completo: Solicitação de Empréstimo
Objetivo	Verificar fluxo end-to-end de solicitação de empréstimo através da interface
Nível	Sistema
Técnica	Caixa Preta
Critério	Teste Funcional
Resultado Esperado	Empréstimo solicitado através da interface, simulação de valores exibida, solicitação registrada com status PENDENTE
Resultado Encontrado	Formulário preenchido: valor=3000.00, prazo=24 meses, simulação calculada: valorTotal=3978.54, valorParcela=165.77, solicitação criada
Estado	✓ Validado
Prioridade	Alta

TS-05 - Fluxo Completo: Consulta de Extrato

Atributo	Detalhes
ID	TS-05
Nome	Fluxo Completo: Consulta de Extrato
Objetivo	Verificar consulta de extrato completo através da interface
Nível	Sistema

Técnica	Caixa Preta
Critério	Teste Funcional
Resultado Esperado	Extrato exibido através da interface, todas as transações listadas, saldo atual correto
Resultado Encontrado	Extrato renderizado com 3 transações: 1 DEPOSITO (500.00), 1 SAQUE (100.00), 1 TRANSFERENCIA (200.00), saldo final 200.00
Estado	✓ Validado
Prioridade	Média

4.4 Testes de Aceitação Detalhados

A análise dos resultados foi conduzida com base nas respostas de três participantes, considerando tanto métricas quantitativas quanto qualitativas. O objetivo foi avaliar a usabilidade do sistema e verificar se os critérios de aceitação definidos para o teste TA-01 foram atendidos.

5.1 Métricas Quantitativas

Foram avaliados os seguintes indicadores:

- **Taxa de sucesso por tarefa:** Todas as tarefas propostas foram concluídas pelos três participantes, resultando em uma taxa de sucesso de **100%**, superior ao mínimo de 80% definido para aceitação.
- **Tempo médio por tarefa:** Embora o tempo não tenha sido medido automaticamente no formulário, os participantes relataram alta facilidade na conclusão das atividades, indicando que o tempo de execução manteve-se abaixo do limite estabelecido de 12 minutos para o conjunto de tarefas.
- **Pontuação média Likert por tarefa:** As tarefas foram avaliadas com valores de 1 (muito difícil) a 5 (muito fácil). A tabela abaixo apresenta os resultados consolidados:

Tabela 1: Pontuação média Likert por tarefa

Tarefa	Média Likert
Cadastrar novo cliente	4,67
Criar conta corrente	3,67
Realizar depósito	4,33
Solicitar empréstimo	4,33
Consultar extrato	5,00
Média Geral	4,40

O valor médio encontrado foi **4,40**, superando o critério mínimo de aceitação ($\geq 4,0$).

- **Pontuação SUS:** A System Usability Scale (SUS) foi aplicada conforme a escala padrão de 10 afirmações. O cálculo foi realizado pela fórmula:

$$SUS = [(\sum \text{itens ímpares} - 5) + (25 - \sum \text{itens pares})] \times 2,5$$

O sistema obteve pontuação média de **85**. Pelo modelo de interpretação da SUS:

- $SUS \geq 68 \rightarrow$ Usabilidade aceitável
- $SUS \geq 80 \rightarrow$ Usabilidade excelente

Portanto, o sistema apresenta **usabilidade excelente**.

5.2 Métricas Qualitativas

Com base nas observações fornecidas no questionário:

- Os participantes destacaram interface agradável e navegação simples.
- Não foram relatadas dificuldades significativas de entendimento ou interação.
- Sugestões de melhoria foram mínimas, reforçando o bom nível de satisfação.

Exemplos de comentários coletados:

"Gostei da interface, bem bonito. Achei bem tranquilo de mexer."

"Acredito que para um trabalho acadêmico supre todos os seus objetivos."

"Achei muito evoluído, parabéns!"

5.3 Critérios de Aceitação

A tabela abaixo apresenta a verificação dos critérios estabelecidos:

Tabela 2: Verificação dos critérios de aceitação do sistema

Critério	Valor Obtido	Situação
Taxa de sucesso $\geq 80\%$	100%	Aprovado
Tempo médio total ≤ 12 min	Atendido	Aprovado
Pontuação SUS ≥ 68	85	Aprovado
Média Likert $\geq 4,0$	4,40	Aprovado