








Project Name:	didactic-bank-application								
Module Name:	createCurrentAccount								
Reference Document:									
Created by:	Raylander Marques Melo								
Date of creation:	05/12/2022								
Date od review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_CRIAR_CONTA_01	Verificar a criação de conta corrente	Receber uma branch válida, um name válido, um lastName válido, um CPF válido, uma birthday válido e um balance válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Conectar com o banco de dados 3. Salvar a conta corrente no banco de dados 4. Esperar o retorno da conta corrente	<Branch válida> <Name válido> <Lastname válido> <Cpf válido> <Birthday válido> <Balance válido>	Retorno da conta corrente criada			
TC_CRIAR_CONTA_02	Verificar a criação de conta corrente	Receber uma branch inválida, um name válido, um lastName válido, um CPF válido, uma birthday válido e um balance válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Conectar com o banco de dados 3. Salvar a conta corrente no banco de dados 4. Esperar o retorno da conta corrente	<Branch inválida> <Name válido> <Lastname válido> <Cpf válido> <Birthday válido> <Balance válido>	Retorno uma exception.invalid. branch			
TC_CRIAR_CONTA_03	Verificar a criação de conta corrente	Receber uma branch válida, um name inválido, um lastName válido, um CPF válido, uma birthday válido e um balance válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Conectar com o banco de dados 3. Salvar a conta corrente no banco de dados 4. Esperar o retorno da conta corrente	<Branch válida> <Name inválido> <Lastname válido> <Cpf válido> <Birthday válido> <Balance válido>	Retorna erro ao tentar salvar			
TC_CRIAR_CONTA_04	Verificar a criação de conta corrente	Receber uma branch válida, um name válido, um lastName inválido, um CPF válido, uma birthday válido e um balance válido	1. Receber os dados necessários para a execução	1. Receber os parâmetros do método 2. Conectar com o banco de dados 3. Salvar a conta corrente no banco de dados	<Branch válida> <Name válido> <Lastname inválido>	Retorna erro ao tentar salvar			


Project Name:	didactic-bank-application								
Module Name:	login								
Reference Document:									
Created by:	Raylander Marques Melo								
Date of creation:	05/12/2022								
Date od review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_LOGIN_01	Verificar se o login está recebendo e validando os dados corretamente	Receber um username válido e um password válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2.Verificar no banco de dados	<Username válido> <Passwors válido>	Retorna o empregado logado			
TC_LOGIN_02	Verificar se o login está recebendo e validando os dados corretamente	Receber um username inválido e um password válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2.Verificar no banco de dados	<Username inválido> <Passwors válido>	Retorna um exception.inexistent.employee			
TC_LOGIN_03	Verificar se o login está recebendo e validando os dados corretamente	Receber um username válido e um password inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2.Verificar no banco de dados	<Username válido> <Passwors inválido>	Retorna um exception.invalid.password			
TC_LOGIN_04	Verificar se o login está recebendo e validando os dados corretamente	Receber um username inválido e um password inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2.Verificar no banco de dados	<Username inválido> <Passwors inválido>	Retorna um exception.inexistent.employee			
Project Name:	didactic-bank-application								
Module Name:	deposit								
Reference Document:									
Created by:	Raylander Marques Melo								
Date of creation:	05/12/2022								
Date od review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_DEPOSIT_01	Verificar se está depositando corretamente	Receber um operationLocation válido, um branch válido, um accountNumber válido, um envelope válido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2.Verificar no banco de dados a conta 3. Salvar a transferência no banco de dados	<Operation-Location válido> <Branch válido> <AccountNumber válido> <Envelope válido>	Retorna o deposito			


					<Amount válido>				
TC_DEPOSIT_02	Verificar se está depositando corretamente	Receber um operationLocation inválido, um branch válido, um accountNumber válido, um envelope válido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Verificar no banco de dados a conta 3. Salvar a transferência no banco de dados	<Operation- Location inválido> <Branch válido> <AccountNumber válido>	Retorna uma exceção na inserção do banco			
					<Envelope válido> <Amount válido>				
TC_DEPOSIT_03	Verificar se está depositando corretamente	Receber um operationLocation válido, um branch inválido, um accountNumber válido, um envelope válido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Verificar no banco de dados a conta 3. Salvar a transferência no banco de dados	<Operation- Location válido> <Branch inválido> <AccountNumber válido>	Retorna uma exceção na validação do banco			
					<Envelope válido> <Amount válido>				
TC_DEPOSIT_04	Verificar se está depositando corretamente	Receber um operationLocation válido, um branch válido, um accountNumber inválido, um envelope válido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Verificar no banco de dados a conta 3. Salvar a transferência no banco de dados	<Operation- Location válido> <Branch válido> <AccountNumber inválido>	Retorna uma exceção na validação do banco			
					<Envelope válido> <Amount válido>				
TC_DEPOSIT_05	Verificar se está depositando corretamente	Receber um operationLocation válido, um branch válido, um accountNumber válido, um envelope inválido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Verificar no banco de dados a conta 3. Salvar a transferência no banco de dados	<Operation- Location válido> <Branch válido> <AccountNumber válido>	Retorna uma exceção na inserção do banco			
					<Envelope inválido> <Amount válido>				
TC_DEPOSIT_06	Verificar se está depositando corretamente	Receber um operationLocation válido, um branch válido, um accountNumber válido, um envelope válido e um amount inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Verificar no banco de dados a conta 3. Salvar a transferência no banco de dados	<Operation- Location válido> <Branch válido> <AccountNumber válido>	Retorna uma exceção na inserção do banco			
					<Envelope válido> <Amount inválido>				

TC_DEPOSIT_07	Verificar se está depositando corretamente	Receber um operationLocation inválido, um branch inválido, um accountNumber inválido, um envelope inválido e um amount inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Verificar no banco de dados a conta 3. Salvar a transferência no banco de dados	<Operation-Location inválido> <Branch inválido> <AccountNumber inválido> <Envelope inválido> <Amount inválido>	Retorna uma exceção na validação do banco			
Project Name:	didactic-bank-application								
Module Name:	getBalance								
Reference Document:									
Created by:	Raylander Marques Melo								
Date of creation:	05/12/2022								
Date of review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_GET_SALDO_01	Verificar se retorna o saldo correto	Receber um branch válida e um accountNumber válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o saldo	<Branch válido> <Passwords válido>	Retorna o saldo da conta			
TC_GET_SALDO_02	Verificar se retorna o saldo correto	Receber um branch inválida e um accountNumber válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o saldo	<Branch inválido> <Passwords válido>	Retorna uma exceção			
TC_GET_SALDO_03	Verificar se retorna o saldo correto	Receber um branch válida e um accountNumber inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o saldo	<Branch válido> <Passwords inválido>	Retorna uma exceção			
TC_GET_SALDO_04	Verificar se retorna o saldo correto	Receber um branch inválida e um accountNumber inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o saldo	<Branch inválido> <Passwords inválido>	Retorna uma exceção			
Project Name:	didactic-bank-application								
Module Name:	getOperationLocation								
Reference Document:									
Created by:	Raylander Marques Melo								
Date of creation:	05/12/2022								

Date od review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_GET_OPERATION_LOCATION_01	Verificar se retorna a operação correta	Receber um operationLocationNumber válida	1. Receber os dados necessários para a execução do método	1. Receber o parâmetros do método 2. Pedir o retorno da localização da operação para o banco de dados	<Operation-Location-Number válido>	Retorna a localização da operação			
TC_GET_OPERATION_LOCATION_02	Verificar se retorna a operação correta	Receber um operationLocationNumber inválida	1. Receber os dados necessários para a execução do método	1. Receber o parâmetros do método 2. Pedir o retorno da localização da operação para o banco de dados	<Operation-Location-Number inválido>	Retorna uma exceção na pesquisa do banco			
Project Name:	didactic-bank-application								
Module Name:	getStatementByDate								
Reference Document:									
Created by:	Raylander Marques Melo								
Date of creation:	05/12/2022								
Date od review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_GET_STATEMENT_BY_DATE_01	Verificar se o extrato por da data está correto	Receber um currentAccount válida, um begin válido e um end válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna os extratos 3. Relizar um sort na lista de extratos	<CurrentAccount válido> <Begin válido> <End válido>	Retorna o extrato por data			
TC_GET_STATEMENT_BY_DATE_02	Verificar se o extrato por da data está correto	Receber um currentAccount inválida, um begin válido e um end válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna os extratos 3. Relizar um sort na lista de extratos	<CurrentAccount inválido> <Begin válido> <End válido>	Retorna exceção			
TC_GET_STATEMENT_BY_DATE_03	Verificar se o extrato por da data está correto	Receber um currentAccount válida, um begin inválido e um end válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna os extratos 3. Relizar um sort na lista de extratos	<Current ccount válido> <Begin inválido> <End válido>	Retorna exceção			
		Receber um	1. Receber os	1. Receber os parâmetros do método	<Current ccount válido>				

TC_GET_STATEMENT_BY_DATE_04	Verificar se o extrato por da data está correto	currentAccount válida, um begin válido e um end inválido	1. Receber os dados necessários para a execução do método	2. Chamar o método que retorna os extratos 3. Relizar um sort na lista de extratos	<Begin válido> <End inválido>	Retorna exceção			
TC_GET_STATEMENT_BY_DATE_05	Verificar se o extrato por da data está correto	Receber um currentAccount inválida, um begin inválido e um end inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna os extratos 3. Relizar um sort na lista de extratos	<Current ccount inválido> <Begin inválido> <End inválido>	Retorna exceção			
Project Name:	didactic-bank-application								
Module Name:	getStatementByMonth								
Reference Document:									
Created by:	Raylander Marques Melo								
Date of creation:	05/12/2022								
Date od review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_GET_STATEMENT_BY_MONTH_01	Verificar se o extrato a por mês está correto	Receber um branch válida, um accountNumber válido, um month válido e um year válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o extrato	<Branch válido> <Account Number válido> <Month válido> <Year válido>	Retorna o extrato por por mês			
TC_GET_STATEMENT_BY_MONTH_02	Verificar se o extrato a por mês está correto	Receber um branch inválida, um accountNumber válido, um month válido e um year válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o extrato	<Branch inválido> <Account Number válido> <Month válido> <Year válido>	Retorna exceção			
TC_GET_STATEMENT_BY_MONTH_03	Verificar se o extrato a por mês está correto	Receber um branch válida, um accountNumber inválido, um month válido e um year válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o extrato	<Branch válido> <Account Number inválido> <Month válido> <Year válido>	Retorna exceção			
TC_GET_STATEMENT_BY_MONTH_04	Verificar se o extrato a por mês está correto	Receber um branch válida, um accountNumber válido, um month inválido e um year válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o extrato	<Branch válido> <Account Number válido> <Month inválido> <Year válido>	Retorna exceção			
TC_GET_STATEMENT	Verificar se o extrato a por mês está correto	Receber um branch válida, um accountNumber válido, um month	1. Receber os dados necessários	1. Receber os parâmetros do método 2. Chamar o método que retorna o extrato	<Branch válido> <Account Number válido>	Retorna exceção			

_BY_MONTH_05	mes esta correto	valido, um month válido e um year inválido	para a execução do método		<Month válido> <Year inválido> <Branch inválido> <Account Number inválido> <Month inválido> <Year inválido>	Retorna exceção			
TC_GET_STATEMENT _BY_MONTH_06	Verificar se o extrato a por mês está correto	Receber um branch inválida, um accountNumber inválido, um month inválido e um year inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Chamar o método que retorna o extrato		Retorna exceção			
Project Name:	didactic-bank-application								
Module Name:	readCurrentAccount								
Reference Document:									
Created by:	Raylander Marques Melo								
Date of creation:	05/12/2022								
Date od review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_RAED_CURRENT _ACCOUNT_01	Verificar se a leitura da conta corrente está correta	Receber um branch válida e um accountNumber válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Realizar a pesquisa no banco de dados	<Branch válido> <Account Number válido>	Retorna a leitura da conta corrente			
TC_RAED_CURRENT _ACCOUNT_02	Verificar se a leitura da conta corrente está correta	Receber um branch inválida e um accountNumber válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Realizar a pesquisa no banco de dados	<Branch inválido> <Account Number válido>	Retorna uma exceção			
TC_RAED_CURRENT _ACCOUNT_03	Verificar se a leitura da conta corrente está correta	Receber um branch válida e um accountNumber inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Realizar a pesquisa no banco de dados	<Branch válido> <Account Number inválido>	Retorna uma exception.inexistent. account			
TC_RAED_CURRENT _ACCOUNT_04	Verificar se a leitura da conta corrente está correta	Receber um branch inválida e um accountNumber inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método 2. Realizar a pesquisa no banco de dados	<Branch inválido> <Account Number inválido>	Retorna uma exceção			
Project Name:	didactic-bank-application								
Module Name:	transfer								
Reference Document:									
Created by:	Raylander Marques Melo								

Date of creation:	05/12/2022								
Date od review:									
TESTE CASE ID	TESTE SCENARIO	TESTE CASE	PRE-CONDITION	TESTE STEPS	TESTE DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS(PASS/FAIL)
TC_TRANSFER_01	Verificar se a tranferencia foi realizada corretamente	Receber um operationLocation válido, um srcBranch válido, um srcAccountNumber válido, um dstBranch válido, um dstAccountNumber válido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location válido>	Retorna a transferência			
				2.Enviar os dados para o método readCurrentAccount	<SrcBranch válido>				
				3. Esperar o retorno da transferência	<Src-Account-Number válido>				
					<DstBranch válido>				
					<Dst-Account-Number válido>				
TC_TRANSFER_02	Verificar se a tranferencia foi realizada corretamente	Receber um operationLocation inválido, um srcBranch válido, um srcAccountNumber válido, um dstBranch válido, um dstAccountNumber válido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location inválido>	Retorna uma exceção			
				2.Enviar os dados para o método readCurrentAccount	<SrcBranch válido>				
				3. Esperar o retorno da transferência	<Src-Account-Number válido>				
					<DstBranch válido>				
					<Dst-Account-Number válido>				
TC_TRANSFER_03	Verificar se a tranferencia foi realizada corretamente	Receber um operationLocation válido, um srcBranch inválido, um srcAccountNumber válido, um dstBranch válido, um dstAccountNumber válido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location válido>	Retorna uma exceção			
				2.Enviar os dados para o método readCurrentAccount	<SrcBranch inválido>				
				3. Esperar o retorno da transferência	<Src-Account-Number válido>				
					<DstBranch válido>				
					<Dst-Account-Number válido>				
				1. Receber os parâmetros do método	<Operation-Location válido>				

TC_TRANSFER_04	Verificar se a transferencia foi realizada corretamente	Receber um operationLocation válido, um srcBranch válido, um srcAccountNumber inválido, um dstBranch válido, um dstAccountNumber válido e um amount válido	1. Receber os dados necessários para a execução do método	2. Enviar os dados para o método readCurrentAccount	<SrcBranch válido>	Retorna uma exceção			
				3. Esperar o retorno da transferência	<Src-Account-Number inválido>				
					<DstBranch válido>				
					<Dst-Account-Number válido>				
					<Amount válido>				
TC_TRANSFER_05	Verificar se a transferencia foi realizada corretamente	Receber um operationLocation válido, um srcBranch válido, um srcAccountNumber válido, um dstBranch inválido, um dstAccountNumber válido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location válido>	Retorna uma exceção			
				2. Enviar os dados para o método readCurrentAccount	<SrcBranch válido>				
				3. Esperar o retorno da transferência	<Src-Account-Number válido>				
					<DstBranch inválido>				
					<Dst-Account-Number válido>				
TC_TRANSFER_06	Verificar se a transferencia foi realizada corretamente	Receber um operationLocation válido, um srcBranch válido, um srcAccountNumber válido, um dstBranch válido, um dstAccountNumber inválido e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location válido>	Retorna uma exceção			
				2. Enviar os dados para o método readCurrentAccount	<SrcBranch válido>				
				3. Esperar o retorno da transferência	<Src-Account-Number válido>				
					<DstBranch válido>				
					<Dst-Account-Number inválido>				
TC_TRANSFER_07	Verificar se a transferencia foi realizada corretamente	Receber um operationLocation válido, um srcBranch válido, um srcAccountNumber válido, um dstBranch válido	1. Receber os dados necessários para a execução	1. Receber os parâmetros do método	<Operation-Location válido>	Retorna uma exceção			
				2. Enviar os dados para o método readCurrentAccount	<SrcBranch válido>				
				3. Esperar o retorno da transferência	<Src-Account-Number válido>				
					<DstBranch válido>				
					<Dst-Account-Number inválido>				

TC_WITHDRAWAL_03	Verificar se cancelamento está cancelando corretamente	Receber um operationLocation válida, uma branch inválida, um accountNumber válida e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location válido>	Retorna uma exceção			
				2. Enviar os dados para o método readCurrentAccount	<Account Number válido>				
				3. Realizar o cancelamento	<Branch inválido>				
					<Amount válido>				
TC_WITHDRAWAL_04	Verificar se cancelamento está cancelando corretamente	Receber um operationLocation válida, uma branch válida, um accountNumber inválida e um amount válido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location válido>	Retorna uma exceção			
				2. Enviar os dados para o método readCurrentAccount	<Account Number inválido>				
				3. Realizar o cancelamento	<Branch válido>				
					<Amount válido>				
TC_WITHDRAWAL_05	Verificar se cancelamento está cancelando corretamente	Receber um operationLocation válida, uma branch válida, um accountNumber válida e um amount inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location válido>	Retorna uma exceção			
				2. Enviar os dados para o método readCurrentAccount	<Account Number válido>				
				3. Realizar o cancelamento	<Branch válido>				
					<Amount inválido>				
TC_WITHDRAWAL_06	Verificar se cancelamento está cancelando corretamente	Receber um operationLocation inválida, uma branch inválida, um accountNumber inválida e um amount inválido	1. Receber os dados necessários para a execução do método	1. Receber os parâmetros do método	<Operation-Location inválido>	Retorna uma exceção			
				2. Enviar os dados para o método readCurrentAccount	<Account Number inválido>				
				3. Realizar o cancelamento	<Branch inválido>				
					<Amount inválido>				