

Plano de Teste

API - Sincredi

1 - Resumo

Este documento descreve os requisitos a testar, os tipos de testes definidos para cada iteração e os software a serem empregados.

Com esse documento devemos::

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os Requisitos a testar.
- Recomendar e descrever as estratégias de teste a serem empregadas.
- Identificar os recursos necessários e prover uma estimativa dos esforços de teste.
- Listar os elementos resultantes do projeto de testes.

2 - Requisitos a Testar

Esta seção deve conter os requisitos a serem testados, assim como as condições para as funcionalidades

As funções serão descritas no quadro a seguir, os tributos e regras da API está abaixo

Atributo	Obrigatório?	Regra
cpf	sim	texto informando o CPF não no formato 999.999.999-99
nome	sim	texto informando o nome da pessoa
email	sim	texto informado um e-mail válido
valor	sim	valor da simulação que deve ser igual ou maior que R\$ 1.000 e menor ou igual que R\$ 40.000
parcela	sim	número de parcelas para pagamento que deve ser igual ou maior que 2 e menor ou igual a 48
seguro	sim	booleano true se com seguro e false se

		sem seguro
--	--	------------

Requisito:

Requisito (Função)			
Funções	Descrição	URI	Resultados
Restrições	Tem a finalidade de consultar o CPF informando, retornando se ele possui ou não uma restrição.	GET <host>/api/v1/restricoes/{cpf}	<ul style="list-style-type: none"> Se não possui restrição do HTTP Status 204 é retornado Se possui restrição o HTTP Status 200 é retornado com a mensagem "O CPF 999999999999 possui restrição"
Consultar todas as simulações cadastradas	Tem a finalidade de listar as simulações cadastradas	GET <host>/api/v1/simulacoes	<ul style="list-style-type: none"> Retorna STATUS 200 e a lista de simulações cadastradas e existir uma ou mais Retorna HTTP Status 204 se não existir simulações cadastradas
Consultar uma simulação pelo CPF	Tem a finalidade de retornar uma simulação previamente cadastrada do CPF indicado	GET <host>/api/v1/simulacoes/{cpf}	<ul style="list-style-type: none"> Retorna a simulação cadastrada Se o CPF não possuir uma simulação o HTTP Status 404 é retornado
Criar uma simulação	Tem a finalidade de inserir uma nova simulação	POST <host>/api/v1/simulacoes	<ul style="list-style-type: none"> Uma simulação cadastrada com sucesso retorna o HTTP Status 201 e os dados inseridos como retorno Uma simulação com problema em alguma regra retorna o HTTP Status 400 com a lista de erros

			<ul style="list-style-type: none"> Uma simulação para um mesmo CPF retorna um HTTP Status 409 com a mensagem "CPF já existente"
Alterar uma simulação	Tem finalidade de alterar uma simulação já existente	PUT <host>/api/v1/simulacoes/{cpf}	<ul style="list-style-type: none"> A alteração pode ser feita em qualquer atributo da simulação As mesmas regras se mantêm Se o CPF não possuir uma simulação o HTTP Status 404 é retornado com a mensagem "CPF não encontrado"
Remover uma simulação	Tem a finalidade de remover uma simulação cadastrada	DELETE <host>/api/v1/simulacoes/{id}	<ul style="list-style-type: none"> Retorna o HTTP Status 204 se simulação for removida com sucesso Retorna o HTTP Status 404 com a mensagem "Simulação não encontrada" se não existir a simulação pelo ID informado

3 - Tipos de teste

Esta seção deve conter os tipos de testes escolhidos para o projeto. Com base no guia de testes, indique os tipos de testes que melhor se adequam aos requisitos, caso necessário complemente ou forneça mais detalhes da técnica e dos critérios de completude sugeridos no guia para cada tipo de teste indicado.

- Testes de integração;
- Testes de aceitação;
- Entre outros.
- Os Tipos de testes, estarão sublocados nos testes do tópico 3.1

3.1 -

Projeto:	Sincredi	ID do caso de teste:	ID - 1
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Validar Se o CPF tem restrição		
Pré-Condições			
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um GET com o curl <host>/api/v1/restricoes/{cpf}		
	4 Caso o CPF tenha Restrição, Deve ser retornado Status Code 200		
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 2
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Validar Se o CPF não tem restrição		
Pré-Condições			
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um GET com o curl <host>/api/v1/restricoes/{cpf}		
	4 Caso o CPF tenha Restrição, Deve ser retornado Status Code 204		
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 3
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Verificar Solicitações Existentes		
Pré-Condições	Validar Simulações existentes		
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um GET com o curl <host>/api/v1/simulacoes		
	4 Será retornado STATUS 200 e em JSON as simulações existentes		
	5 Caso Não tenha simulações Deve ser retornado o status 204		
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 4
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Validar simulação existente por CPF		
Pré-Condições			
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um GET com o curl <host>/api/v1/simulacoes/{CPF}		
	4 Caso exista Simulação para o CPF. Será retornado STATUS 200 e em json a simulação		
	5 Caso não tenha simulações deve ser retornado 404		
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 5
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Validar uma alteração de uma simulação		
Pré-Condições			
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um PUT com o curl <host>/api/v1/simulacoes/{cpf}		
	o PUT deve ser enviado em JSON com a informação da simulação, e o dado a ser atualizado: { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, "parcelas": INT , "seguro": BOOLEAN 4 }		
	5 Caso exista Exista o CPF. Será atualizado os dados e STATUS 200		
		Caso Não exista o CPF será retornado status 404 Indicando que o CPF não foi encontrado	
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 6
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Validar deleção de uma simulação		
Pré-Condições	Saber o ID da Simulação		
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um DELETE com o curl <host>/api/v1/simulacoes/{ID}		
	4 Caso exista Exista o ID . Será deletado a simulação e STATUS 204		
	5 Caso Não exista o ID será retornado status 404 Indicando que a Simulação não foi encontrado		

Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 7
Ambiente:	Insomnia		
Autor:	Yuri		
Titulo:	Criar uma simulação com dados Corretos e dentro das condições		
Pré-Condições	Cpf, Nome, Email, Valor, Parcela, Seguro		
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um POST com o curl <host>/api/v1/simulacoes/{cpf}		
	o POST deve ser enviado em JSON com a informação da simulação { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, (Deve estar entre Mil reais até 40 mil reais) "parcelas": INT , (Deve estar entre 2 a 48 Parcelas) "seguro": BOOLEAN 4 }		
	POST enviado { "id": 1, "nome": "Fulano", "cpf": "66414919004", "email": "fulano@gmail.com", "valor": 11000.00, "parcelas": 3, "seguro": true 5 },		
	6 Deve ser retornado que a simulação foi cadastrada com sucesso e o retorno 201		
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 8
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Criar uma simulação com CPF já existente		
Pré-Condições			
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um POST com o curl <host>/api/v1/simulacoes/{cpf}		
	o POST deve ser enviado em JSON com a informação da simulação { "nome": " STRING ", "cpf": INT , "email": " STRING ", "valor": INT , (Deve estar entre Mil reais até 40 mil reais) "parcelas": INT , (Deve estar entre 2 a 48 Parcelas) "seguro": BOOLEAN }		
	POST enviado { "id": 1, "nome": "Fulano", "cpf": "66414919004", "email": "fulano@gmail.com", "valor": 11000.00, "parcelas": 3, "seguro": true },		
	Deve ser retornado que o CPF já existe e o retorno 409		
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 9
Ambiente:	Insomnia		
Autor:	Yuri		
Titulo:	Criar uma simulação com dados fora da condição(Valor)		
Pré-Condições	Cpf, Nome, Email, Valor, Parcela, Seguro		
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um POST com o curl <host>/api/v1/simulacoes/{cpf}		
	o POST deve ser enviado em JSON com a informação da simulação { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, (Deve estar entre Mil reais até 40 mil reais) "parcelas": INT , (Deve estar entre 2 a 48 Parcelas) "seguro": BOOLEAN 4 }		
	5 POST enviado { "id": 1, "nome": "beltrado", "cpf": "66414919012", "email": "fulano1@gmail.com", "valor": 900.00, (VALOR ERRADO) "parcelas": 3, "seguro": true },		
	6 Deve ser retornado que existe uma problema na simulação Pelo valor ser menor que 1000, indicar o erro e o retorno 400		
	Deve ser retornado que existe uma problema na simulação Pelo valor ser maior que 40000, indicar o erro e o retorno 400		
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 10
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Criar uma simulação com dados fora da condição(Parcela)		
Pré-Condições	Cpf, Nome, Email, Valor, Parcela, Seguro		
Passos:	Descrição	Status	
	1 Montar Ambiente API		
	2 Definir o URI <HOST> =http://localhost:8080		
	3 Enviar um POST com o curl <host>/api/v1/simulacoes/{cpf}		
	o POST deve ser enviado em JSON com a informação da simulação { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, (Deve estar entre Mil reais até 40 mil reais) "parcelas": INT , (Deve estar entre 2 a 48 Parcelas) "seguro": BOOLEAN 4 }		
	5 POST enviado { "id": 11, "nome": "Fulano", "cpf": "66414919111", "email": "fulano@gmail.com", "valor": 11000.00, "parcelas": 1, (Parcela INVALIDA) "seguro": true }		
	6 Deve ser retornado que existe uma problema na simulação pela parcela ser menor que 2, indicar o erro e o retorno 400		
	Deve ser retornado que existe uma problema na simulação pela parcela ser maior que 2, indicar o erro e o retorno 400		
Pós condições (Resultado final)			

Projeto:	Sincredi	ID do caso de teste:	ID - 11
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Criar uma simulação com dados fora da condição(Todos)		
Pré-Condições	Cpf, Nome, Email, Valor, Parcela, Seguro		
Passos:	Descrição	Status	
1	Montar Ambiente API		
2	Definir o URI <HOST> =http://localhost:8080		
3	Enviar um POST com o curl <host>/api/v1/simulacoes/{cpf}		
4	<p>o POST deve ser enviado em JSON com a informação da simulação {</p> <pre> "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, (Deve estar entre Mil reais até 40 mil reais) "parcelas": INT , (Deve estar entre 2 a 48 Parcelas) "seguro": BOOLEAN </pre>		
5	<pre> POST enviado{ "id": 123, "nome": 123, "cpf": 123, "email": 1, "valor": 920.00, "parcelas": "2", "seguro": 4 } </pre>		
6	Deve ser retornado que existe uma problema na simulação, indicar o erro e o retorno 400		
Pós condições (Resultado final)			

4 - Recursos

Esta seção deve descrever os recursos humanos, de ambiente de teste (hardware e software) e de ferramentas de automatização de testes necessários para execução dos testes que devem ser descritos nas subseções que seguem.

4.1 - Ambiente de teste - Software e Hardware

Descreva aqui o hardware e sua configuração, e o software. Por exemplo, o sistema operacional, browsers, servidor web, etc.

4.2 - Ferramenta de teste

Equipamento:

- Computador com pelo menos 4 GB de RAM
- Sistema operacional Windows, Android, MacOS, Linux
- Navegador web Google Chrome ou Firefox

Software:

- API de simulação de crédito versão 1.0
- Ferramenta de teste de API, como Postman

Dados:

- Dados para testes usado do site 4DEVs, para testes validos e invalidos

Procedimentos:

- Deve ser seguido os testes de API conforme documentação, com as regras definidas e o escopo do caso de teste

Resultados:

- Deve ser indicado no casos de testes os casos validados com OK, e os casos que não for conforme esperado com FALHA

Conclusão:

- Ao fim do documento, deve-se analisar se tem o escopo suficiente e segue a regra de negocio

5 - Criterio de aceitação

Todos os casos de testes devem passar, aqueles que derem FALHA, deve retornar ao desenvolvimento, ser ajustado, e testado novamente o plano de teste POR COMPLETO.

6 - Cronograma de teste

- Planejamento para plano de teste: A Definir
- Execução dos testes: A Definir
- Registro e relatório de bugs encontrados: Imediato