

Projeto:	Sincroni	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	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um GET com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	Caso o CPF tenha Restrição, Deve ser retornado Status Code 200	OK	

Pós condições (Resultado final)

```
1 {
2   "mensagem": "O CPF 97093236014 tem problema"
3 }
```

Projeto:	Sincroni	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	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um PUT com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	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 }	OK	
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	OK	

Pós condições (Resultado final)

```
1 {
2   "mensagem": "CPF 6644919004 não encontrado"
3 }
```

Projeto:	Sincroni	ID do caso de teste:	ID - 9
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Criar uma simulação com dados fora da condição(Valor)		
Pré-Condições	Cpf.Nome,Email,Valor.Parcela.Seguro		
Passos:	Descrição	Status	
1	Montar Ambiente API	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um POST com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	o POST deve ser enviado em JSON com a informação da simulação { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, "parcelas": INT, "seguro": BOOLEAN }	OK	
5	POST enviado { "cpf": 1, "nome": "fulano", "email": "fulano@gmail.com", "valor": 900.00, "parcelas": 3, "seguro": true }	OK	
6	Deve ser retornado que existe uma problema na simulação Pelo valor ser menor que 1000, indicar o erro e o retorno 400	Falha	
7	Deve ser retornado que existe uma problema na simulação Pelo valor ser maior que 40000, indicar o erro e o retorno 400	OK	

Pós condições (Resultado final)

```
1 {
2   "error": {
3     "mensagem": "valor: 900.00, (VALOR ERRADO)"
4   }
5 }
```

Projeto:	Sincroni	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	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um GET com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	Caso o CPF tenha Restrição, Deve ser retornado Status Code 204	OK	

Pós condições (Resultado final)

No body returned for response

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

Pós condições (Resultado final)

OK

Projeto:	Sincroni	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	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um POST com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	o POST deve ser enviado em JSON com a informação da simulação { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, "parcelas": INT, "seguro": BOOLEAN }	OK	
5	POST enviado { "cpf": 11, "nome": "fulano", "email": "fulano@gmail.com", "valor": 11000.00, "parcelas": 1, "seguro": true }	OK	
6	Deve ser retornado que existe uma problema na simulação pela parcela ser menor que 2, indicar o erro e o retorno 400	OK	
7	Deve ser retornado que existe uma problema na simulação pela parcela ser maior que 2, indicar o erro e o retorno 400	Falha	

Pós condições (Resultado final)

```
1 {
2   "error": {
3     "mensagem": "parcela: 1, (Parcela INVALIDA)"
4   }
5 }
```

Projeto:	Sincroni	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	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um GET com o curl <host>+api/v1/simulacoes	OK	
4	Será retornado STATUS 200 e em JSON as simulações existentes	OK	
5	Caso Não tenha simulação Deve ser retornado o status 204	Falha	

Pós condições (Resultado final)

```
1 {
2   "simulacoes": [
3     {
4       "cpf": 1,
5       "nome": "fulano",
6       "email": "fulano@gmail.com",
7       "valor": 11000.00,
8       "parcelas": 3,
9       "seguro": true
10    }
11  ]
12 }
```

Projeto:	Sincroni	ID do caso de teste:	ID - 7
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	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	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um POST com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	o POST deve ser enviado em JSON com a informação da simulação { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, "parcelas": INT, "seguro": BOOLEAN }	OK	
5	POST enviado { "cpf": 1, "nome": "fulano", "email": "fulano@gmail.com", "valor": 11000.00, "parcelas": 3, "seguro": true }	OK	
6	Deve ser retornado que a simulação foi cadastrada com sucesso e o retorno 201	OK	

Pós condições (Resultado final)

```
1 {
2   "cpf": 1,
3   "nome": "fulano",
4   "email": "fulano@gmail.com",
5   "valor": 11000.00,
6   "parcelas": 3,
7   "seguro": true
8 }
```

Projeto:	Sincroni	ID do caso de teste:	ID - 8
Ambiente:	Insomnia		
Autor:	Yuri		
Título:	Criar uma simulação com CPF já existente		
Pré-Condições	Cpf.Nome,Email,Valor.Parcela.Seguro		
Passos:	Descrição	Status	
1	Montar Ambiente API	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um POST com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	o POST deve ser enviado em JSON com a informação da simulação { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, "parcelas": INT, "seguro": BOOLEAN }	OK	
5	POST enviado { "cpf": 123, "nome": 123, "email": 1, "valor": 920.00, "parcelas": 2, "seguro": 4 }	OK	
6	Deve ser retornado que existe uma problema na simulação, indicar o erro e o retorno 400	Falha	

Pós condições (Resultado final)

```
1 {
2   "error": {
3     "mensagem": "valor: 920.00, (VALOR ERRADO)"
4   }
5 }
```

Embora todos os campos estivessem incorretos, notifiquei somente o problema no EMAIL, com email válido e o resto incorreto, a simulação é criada

Projeto:	Sincroni	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	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um GET com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	Caso exista Simulação para o CPF. Será retornado STATUS 200 e em json a simulação	OK	
5	Caso não tenha simulações deve ser retornado 404	OK	

Pós condições (Resultado final)

```
1 {
2   "cpf": 1,
3   "nome": "fulano",
4   "email": "fulano@gmail.com",
5   "valor": 11000.00,
6   "parcelas": 3,
7   "seguro": true
8 }
```

Projeto:	Sincroni	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	OK	
2	Definir o URI <HOST> >http://localhost:8080	OK	
3	Enviar um POST com o curl <host>+api/v1/simulacoes/<cpf>	OK	
4	o POST deve ser enviado em JSON com a informação da simulação { "nome": "STRING", "cpf": INT, "email": "STRING", "valor": INT, "parcelas": INT, "seguro": BOOLEAN }	OK	
5	POST enviado { "cpf": 1, "nome": "fulano", "email": "fulano@gmail.com", "valor": 11000.00, "parcelas": 3, "seguro": true }	OK	
6	Deve ser retornado que o CPF já existe e o retorno 409	Falha	

Pós condições (Resultado final)

```
1 {
2   "mensagem": "CPF duplicado"
3 }
```