



Avaliação Prática

Implemente os testes unitários referente aos serviços da API Aluno, conforme modelo abaixo:

Teste 01: Inserir Aluno com CPF **válido**

Exemplo:

Nome do Método: saveAluno(String requestBodyJSON)	
Envio para a API Aluno	Retorno da API Aluno
{ "id":"2024123456_001", "nome": "Benevaldo Gonçalves", "cpf": "57014540297", "email": "benevaldo@gmail.com" }	{ "id": "2024123456_001", "status": 1, "message": "Aluno incluído com sucesso." }

Teste 02: Deve invalidar a inclusão de aluno com o mesmo **Id**

Exemplo:

Nome do Método: saveAluno(String requestBodyJSON)	
Envio para a API Aluno	Retorno da API Aluno
{ "id":"2024123456_001", "nome": "Benevaldo Gonçalves", "cpf": "57014540297", "email": "benevaldo@gmail.com" }	{ "id": "2024123456_001", "status": 3, "message": "Aluno já cadastrado." }

Teste 03: Invalidar a inclusão de um **NOVO** Aluno com CPF **Inválido**.

Exemplo:

Nome do Método: saveAluno(String requestBodyJSON)	
Envio para a API Aluno	Retorno da API Aluno
{ "id":"2024123456_002", "nome": "Bruno Pereira", "cpf": "12345678901", "email": "bruno@gmail.com" }	{ "id": "2024123456_002", "status": 3, "message": "Falha na inclusão: CPF inválido." }

Teste 04: Deve invalidar na inclusão, um **NOVO** aluno com CPF **válido** com os campos **nulos**: **nome** ou **email**.

Exemplo:

Nome do Método: saveAluno(String requestBodyJSON)	
Envio para a API Aluno	Retorno da API Aluno
{ "id": "2024123456_002", "nome": null, "cpf": "57014540297", "email": "teste@gmail.com" }	{ "id": "2024123456_002", "status": 3, "message": "Os campos nome ou email são obrigatórios." }

Teste 05: Deve obter Aluno **existente** pelo **Id**

Exemplo:

Nome do Método: getAlunoById(String id)	
Envio para a API Aluno	Retorno da API Aluno
2024123456_001	{ "id": "2024123456_001", "nome": "Benevaldo Gonçalves", "cpf": "57014540297", "email": "benevaldo@gmail.com" }

Teste 06: Deve verificar Aluno **inexistente** pelo **Id**

Exemplo:

Nome do Método: getAlunoById(String id)	
Envio para a API Aluno	Retorno da API Aluno
2024123456_200	{ "id": null, "nome": null, "cpf": null, "email": null }

Teste 07: Deve **atualizar** aluno **existente** com CPF **válido**.

Exemplo:

Nome do Método: updateAluno(String requestBodyJSON)	
Envio para a API Aluno	Retorno da API Aluno
{ "id": "2024123456_001", "nome": "Nicolas Gonçalves", "cpf": "57014540297", "email": "nicolas@gmail.com" }	{ "id": "2024123456_001", "status": 1, "message": "Aluno alterado com sucesso." }

Teste 08: Deve invalidar **atualização** de aluno **inexistente** com CPF **válido**

Exemplo:

Nome do Método: updateAluno(String requestBodyJSON)	
Envio para a API Aluno	Retorno da API Aluno
{ "id": "2024123456_200", "nome": "Maria Clara", "cpf": "57014540297", "email": "maria@gmail.com" }	{ "id": "2024123456_200", "status": 3, "message": "Falha na alteração: aluno não existe." }

Teste 09: Deve invalidar **exclusão** de aluno **inexistente**

Exemplo:

Nome do Método: deleteAluno(String id)	
Envio para a API Aluno	Retorno da API Aluno
2024123456_200	{ "id": "2024123456_200", "status": 3, "message": "Falha na deleção: aluno não existe." }

Teste 10: Deve verificar a **exclusão** do Aluno **existente**

Exemplo:

Nome do Método: deleteAluno(String id)	
Envio para a API Aluno	Retorno da API Aluno
2024123456_001	{ "id": "2024123456_001", "status": 1, "message": "Aluno deletado com sucesso." }