

Faculdade Impacta Tecnologia

Projeto de Software (5º semestre)

Curso: Análise e Desenvolvimento de Sistemas

Turma: 5A - EAD

Matéria: Software Product: Analysis, Specification, Project & Implementation

Professor: Antonio de Oliveira Dias

ALUNOS PARTICIPANTES

Ana Carollyne Guimarães de Souza	RA: 2301929
Luana Ramos de Almeida	RA: 2302018
Natã Pedro	RA: 2301849
Pedro Pescarole	RA: 2301635

SOBRE O PROJETO

Nome: Moveat

Resumo: Esse projeto busca oferecer uma experiência prática para aqueles que praticam atividades físicas. A plataforma permite que pessoais trainers e nutricionistas cadastrem seus usuários e associem planos de treino e alimentação personalizados.

Ao fazer login, os usuários encontram tudo o que precisam para seguir suas rotinas de treino e alimentação. O objetivo da plataforma é trazer praticidade, unindo tudo em um lugar só.

AC 02 – FUNCIONALIDADE DASHBOARD DO NUTRICIONISTA

Objetivo: A segunda sprint do projeto Moveat teve como objetivo a implementação das funcionalidades de registro e leitura de pacientes como um nutricionista autenticado na plataforma.

Link do board do projeto:

<https://github.com/orgs/Moveat-Fit/projects/7>

Link do vídeo apresentando a funcionalidade:

https://www.youtube.com/watch?v=btm__fbYDHk

Link do código-fonte (GitHub):

<https://github.com/orgs/Moveat-Fit/repositories>

I. RELATÓRIO DO TESTE DE API

Responsável QA: Ana Carollyne Guimarães de Souza

Sistema Operacional: Windows 11 Pro

Ferramenta de Teste: Postman (v11.36.3)

Data dos testes: 28/03/2025

1. OBJETIVO DOS TESTES DE API

Validar a funcionalidade dos endpoints da API, verificando as respostas HTTP, a estrutura do JSON das respostas e a segurança da API em relação à autorização de acesso. As validações foram feitas para cenários positivos e também cenários negativos. Todos os testes seguem a mesma estrutura:

```
/* TESTES PARA REGISTRO BEM SUCEDIDO */
if (pm.response.code === 201) {
  pm.test("Status code é 201 - Registro bem sucedido", function() {
    pm.response.to.have.status(201);
  });
  pm.test("Mensagem de sucesso", function() {
    pm.expect(jsonData.message).to.eql("Profissional registrado com sucesso");
  });
  pm.test("Response deve ter a estrutura correta", function() {
    pm.expect(jsonData).to.have.property("message");
    pm.expect(typeof jsonData.message).to.eql("string");
  });
}

/* TESTES PARA CAMPOS OBRIGATÓRIOS AUSENTES */
} else if (pm.response.code === 400) {
  pm.test("Status code é 400 - Campos obrigatórios ausentes", function() {
    pm.response.to.have.status(400);
  });
  pm.test("Mensagem de erro", function() {
    pm.expect(jsonData.message).to.eql("Campos obrigatórios ausentes ou inválidos");
  });
}

/* TESTES PARA EMAIL JÁ CADASTRADO */
} else if (pm.response.code === 409) {
  pm.test("Status code é 409 - Email, CPF ou telefone já cadastrado", function() {
    pm.response.to.have.status(409);
  });
  pm.test("Mensagem de erro", function() {
    pm.expect(jsonData.message).to.eql("Email, CPF ou número de telefone já registrado");
  });
}

/* TESTES PARA ERRO INTERNO DO SERVIDOR */
} else if (pm.response.code === 500) {
  pm.test("Status code é 500 - Erro interno do servidor", function() {
    pm.response.to.have.status(500);
  });
  pm.test("Mensagem de erro", function() {
    pm.expect(jsonData.message).to.include("Erro interno do servidor");
  });
}
}
```

2. ENDPOINTS TESTADOS

Endpoint	Método HTTP	Descrição	Status Esperado
/register	POST	Registra um novo profissional no sistema	201 CREATED
/professional	POST	Realiza o login de um profissional	200 OK
/register/patient	POST	Registra um paciente no sistema (apenas para profissionais autenticados)	201 CREATED
/patient	POST	Realiza o login de um paciente	200 OK
/test-connection	GET	Testa a conexão com o banco de dados	200 OK

3. REGISTRO DE PROFISSIONAL NO SISTEMA (POST /register)

3.1. CENÁRIO POSITIVO: REGISTRO BEM-SUCEDIDO DO NUTRICIONISTA

Status Code esperado	201
Resultado esperado	Mensagem: "Profissional registrado com sucesso"
Resultado obtido	<pre>{ "message": "Profissional registrado com sucesso", "access_token": "<informação-sensível>" }</pre>
Corpo da requisição	<pre>{ "full_name": "Nutricionista de Teste", "email": "teste@nutricionista.com", "password": "Teste123!", "cpf": "12345678900", "phone": "11999999999", "regional_council_type": "CRN", "regional_council": "123456" }</pre>
Evidência	
Implementação	Manual
Status do teste	APROVADO

3.2. CENÁRIO NEGATIVO: TENTATIVA DE REGISTRO COM CAMPOS AUSENTES

Status Code esperado	400
Resultado esperado	Mensagem: “Campos obrigatórios ausentes ou inválidos”
Resultado obtido	{ "message": "Formato de CPF inválido" }
Corpo da requisição	//Campos obrigatórios em branco { "full_name": "", "email": "", "password": "", "cpf": "", "phone": "", "regional_council_type": "", "regional_council": "" }
Evidência	<div><div>PASSED</div>Status code é 400 - Campos obrigatórios ausentes</div> <div><div>FAILED</div>Mensagem de erro AssertionError: expected 'Formato de CPF inválido' to deeply equal 'Campos obrigatórios ausentes ou inválid..'</div>
Implementação	Manual
Status do teste	REPROVADO (28/03/2025) CORRIGIDO E VALIDADO (03/04/2025)

3.3. CENÁRIO NEGATIVO: TENTATIVA DE REGISTRO COM EMAIL CADASTRADO

Status Code esperado	409
Resultado esperado	Mensagem: “Email, CPF ou telefone já registrado”
Resultado obtido	<pre>{ "message": "Email, CPF ou número de telefone já registrado" }</pre>
Corpo da requisição	<pre>{ "full_name": "Nutricionista Email Cadastrado", "email": "teste@nutricionista.com", // Email já cadastrado "password": "Teste123!", "cpf": "66952140685", "phone": "12965262301", "regional_council_type": "CRN", "regional_council": "7854120" }</pre>
Evidência	
Implementação	Manual
Status do teste	APROVADO

4. LOGIN DO PROFISSIONAL NO SISTEMA (POST /professional) | Retorna Token JWT

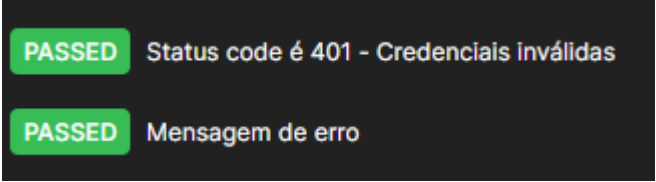
4.1. CENÁRIO POSITIVO: LOGIN BEM-SUCEDIDO DO NUTRICIONISTA

Status Code esperado	200	
Resultado esperado	Response retorna token do nutricionista	
Resultado obtido	{ "access_token": "<informação-sensível>" }	
Corpo da requisição (login com email)	{ "login": "teste@nutricionista.com", "password": "Teste123!" }	
Corpo da requisição (login com CPF)	{ "login": "12345678900", "password": "Teste123!" }	
Corpo da requisição (login com telefone)	{ "login": "11999999999", "password": "Teste123!" }	
Evidência	<div><div>PASSED</div> Status code é 200 - Login bem sucedido</div> <div><div>PASSED</div> Token JWT retornado</div> <div><div>PASSED</div> Response deve ter a estrutura correta</div>	
Implementação	Manual	
Status do teste	APROVADO	

4.2. CENÁRIO NEGATIVO: TENTATIVA DE LOGIN DO NUTRICIONISTA COM CREDENCIAIS INCOMPLETAS

Status Code esperado	400
Resultado esperado	Mensagem: "Email e senha são obrigatórios"
Resultado obtido	<pre>{ "message": "E-mail e senha são obrigatórios" }</pre>
Corpo da requisição	<pre>{ "login": "", "password": "" }</pre>
Evidência	
Implementação	Manual
Status do teste	APROVADO

4.3. CENÁRIO NEGATIVO: TENTATIVA DE LOGIN DO NUTRICIONISTA COM CREDENCIAIS INCORRETAS

Status Code esperado	401
Resultado esperado	Mensagem: “Credenciais inválidas”
Resultado obtido	{ "message": "Credenciais inválidas" }
Corpo da requisição	{ "login": "teste@nutricionista.com", "password": "senhaerrada" }
Evidência	
Implementação	Manual
Status do teste	APROVADO

5. REGISTRO DE PACIENTE NO SISTEMA (POST /register/patient) | Apenas nutricionistas autenticados podem cadastrar novos pacientes)

5.1. CENÁRIO POSITIVO: REGISTRO BEM-SUCEDIDO DO PACIENTE

Status Code esperado	201
Resultado esperado	Mensagem: “Paciente registrado com sucesso”
Resultado obtido	<pre>{ "message": "Paciente registrado com sucesso" }</pre>
Corpo da requisição	<pre>{ "full_name": "Paciente de Teste 2", "birth_date": "2001-09-11", "gender": "F", "email": "paciente2@teste.com", "password": "Teste123!", "mobile": "11996690099", "cpf": "50260030830", "weight": "50", // peso como tipo String "height": "1.50", // altura como tipo String "note": "Diabético" }</pre>
Evidência	<div><div>PASSED</div>Status code é 201 - Paciente registrado com sucesso</div> <div><div>PASSED</div>Response deve conter um ID do paciente</div>
Implementação	Manual
Status do teste	REPROVADO (31/03/2025) CORRIGIDO E VALIDADO (05/04/2025)

5.2. CENÁRIO NEGATIVO: TENTATIVA DE REGISTRO DO PACIENTE COM CAMPOS AUSENTES

Status Code esperado	400	
Resultado esperado	Erros de validação nos campos	
Resultado obtido	<pre>{ "message": "Erros de validação", "errors": { "full_name": "Nome completo deve ter pelo menos 3 caracteres", "email": "Formato de e-mail inválido" } }</pre>	
Corpo da requisição	<pre>{ "full_name": "", "birth_date": "2001-09-11", "gender": "F", "email": "", "password": "Teste123!", "mobile": "11996696709", "cpf": "50262232830", "weight": "50", "height": "1.50", "note": "Nada" }</pre>	
Evidência	<div><div>PASSED</div>Status code é 400 - Erros de validação nos campos</div> <div><div>PASSED</div>Mensagem de erro presente</div> <div><div>PASSED</div>Lista de erros presente</div>	
Implementação	Manual	
Status do teste	APROVADO	

5.3. CENÁRIO NEGATIVO: TENTATIVA DE REGISTRO DO PACIENTE COM EMAIL JÁ CADASTRADO

Status Code esperado	409	
Resultado esperado	Mensagem: “Email, CPF ou telefone já registrado”	
Resultado obtido	{ "message": "Email, CPF ou número de telefone já registrado" }	
Corpo da requisição	{ "full_name": "Paciente de Teste", "birth_date": "2001-09-11", "gender": "M", "email": "paciente2@teste.com", //email cadastrado "password": "Teste123!", "mobile": "11996696709", "cpf": "50262232830", "weight": "50", "height": "1.50", "note": "Nada" }	
Evidência	<div><div>PASSED</div> Status code é 409 - Conflito de dados</div> <div><div>PASSED</div> Mensagem de erro apropriada</div>	
Implementação	Manual	
Status do teste	APROVADO	

5.4. CENÁRIO NEGATIVO: TENTATIVA DE REGISTRO DO PACIENTE COM
ACESSO NÃO AUTORIZADO (NÃO É UM NUTRICIONISTA)

Status Code esperado	403
Resultado esperado	Mensagem: “Acesso não autorizado”
Resultado obtido	{ "message": "Acesso não autorizado" }
Corpo da requisição	// TENTATIVA DE REGISTRO COM TOKEN DE OUTRO PACIENTE { "full_name": "Paciente de Teste 3", "birth_date": "2001-09-11", "gender": "F", "email": "paciente3@hotmail.com", "password": "Teste123!", "mobile": "11994605212", "cpf": "50262232830", "weight": "50", "height": "1.50", "note": "Hipertenso" }
Evidência	<div><div>PASSED</div> Status code é 403 - Acesso negado</div> <div><div>PASSED</div> Mensagem de erro específica</div>
Implementação	Manual
Status do teste	APROVADO

6. LOGIN DO PACIENTE NO SISTEMA (POST /patient) | Retorna Token JWT

6.1. CENÁRIO POSITIVO: LOGIN BEM-SUCEDIDO DO PACIENTE

Status Code esperado	200	
Resultado esperado	Response retorna token do paciente	
Resultado obtido	{ "access_token": "<informação-sensível>" }	
Corpo da requisição (login com email)	{ "login": "paciente@teste.com", "password": "Teste123!" }	
Corpo da requisição (login com CPF)	{ "login": "50262230830", "password": "Teste123!" }	
Corpo da requisição (login com telefone)	{ "login": "11996696799", "password": "Teste123!" }	
Evidência	<div><div>PASSED</div> Status code é 200 - Login bem-sucedido</div> <div><div>PASSED</div> Response deve conter um token JWT</div>	
Implementação	Manual	
Status do teste	APROVADO	

6.2. CENÁRIO NEGATIVO: TENTATIVA DE LOGIN DO PACIENTE COM CREDENCIAIS INCOMPLETAS

Status Code esperado	400
Resultado esperado	Mensagem: “Email e senha são obrigatórios”
Resultado obtido	{ "message": "Login e senha são obrigatórios" }
Corpo da requisição	{ "login": "", "password": "" }
Evidência	<div><div>PASSED</div> Status code é 400 - Credenciais incompletas</div> <div><div>PASSED</div> Mensagem de erro presente</div>
Implementação	Manual
Status do teste	APROVADO

6.3. CENÁRIO NEGATIVO: TENTATIVA DE LOGIN DO PACIENTE COM CREDENCIAIS INCORRETAS

Status Code esperado	401
Resultado esperado	Mensagem: “Credenciais inválidas”
Resultado obtido	{ "message": "Credenciais inválidas" }
Corpo da requisição	{ "login": "paciente@teste.com", "password": "senhaerrada" }
Evidência	<div><div>PASSED</div> Status code é 401 - Credenciais inválidas</div> <div><div>PASSED</div> Mensagem de erro específica</div>
Implementação	Manual
Status do teste	APROVADO

7. **TESTE DE CONEXÃO COM O BANCO (GET /test-connection)**

Status Code esperado	200
Resultado esperado	Mensagem: “Conexão estabelecida com sucesso”
Resultado obtido	<pre>{ "data": [{ "id": 1, "full_name": "Nutricionista de Teste", "email": "teste@nutricionista.com", "password": "pbkdf2:sha256:100000\$9rj4Ds5c\$e79c15dbbe1604e0c2e44d 44d11e97dca6a698880232331e7c0d2d77a9e2618d", "cpf": "12345678900", "phone": "11999999999", "regional_council_type": "CRN", "regional_council": "123456", "created_at": "2025-03-28T21:17:41", "updated_at": "2025-03-28T21:17:41" }, { "id": 5, "full_name": "Nutricionista de Teste 2", "email": "teste2@nutricionista.com", "password": "pbkdf2:sha256:100000\$syJPGrvV\$a784e260dc18af4e029a78 26434d639c27235fd889281c97907896562d4bcd8", "cpf": "12345678901", "phone": "11999999998", "regional_council_type": "CRN", "regional_council": "123436",</pre>

	<pre>"created_at": "2025-03-31T21:44:52", "updated_at": "2025-03-31T21:44:52" }] }</pre>
Evidência	<div><div>PASSED</div>Status code é 200 - Conexão bem sucedida</div> <div><div>FAILED</div>Mensagem de sucesso AssertionError: expected undefined to deeply equal 'Conexão com o backend estabelecida co...'</div>
Implementação	Manual
Status do teste	REPROVADO (28/03/2025) CORRIGIDO E VALIDADO (03/04/2025)

8. BUGS ENCONTRADOS DURANTE OS TESTES DE API

TESTES DE API		
ID	DESCRIÇÃO	STATUS
BUG-API-001	Endpoint /register permite cadastro sem nome	Corrigido
BUG-API-002	Endpoint /register não reconhece campos obrigatórios em branco	Corrigido
BUG-API-003	Endpoint /test-connection retorna array de usuários cadastrados	Corrigido
BUG-API-004	Endpoint /register/patient aceita campos de peso e altura como tipo String	Corrigido
BUG-API-005	Conflito no formato de data entre o MySQL e a API Endpoint /register/patient	Corrigido

II. RELATÓRIO DO TESTE DE SISTEMA (E2E)

Responsável QA: Ana Carollyne Guimarães de Souza

Sistema Operacional: Windows 11 Pro

Data dos testes: 06/04/2025

Relatório completo: [CLIQUE AQUI](#)

9. OBJETIVO

Validar que o sistema permite que o nutricionista realize o cadastro de um novo paciente com sucesso e, em seguida, consiga consultar corretamente as informações desse paciente, garantindo a integridade e disponibilidade dos dados após o registro.

10. PRÉ-CONDIÇÕES

- O nutricionista está autenticado no sistema;
- A funcionalidade do cadastro de paciente está disponível;
- O sistema está operacional;
- Os testes de API já foram concluídos.

11. CASOS DE TESTE EXECUTADOS

Para melhor organização e rastreabilidade, os casos de teste foram criados na plataforma **Qase**. Todos foram executados manualmente.

SPRINT 2: DASHBOARD NUTRICIONISTA (CADASTRO E LEITURA DE USERS) + ✎ 📄 🗑			
+ Create quick test			
Cadastro de Pacientes + ✎ 📄 🗑			
↑	👤	MOVEAT-9	Cadastrar paciente com sucesso
↑	👤	MOVEAT-10	Validar tratamento de erro para cadastro de paciente com CPF inválido
↑	👤	MOVEAT-11	Validar tratamento de erro para cadastro de paciente com e-mail inválido
↑	👤	MOVEAT-12	Validar tratamento de erro para cadastro de paciente sem campos obrigatórios preenchidos
+ Create quick test			
Pesquisa (leitura) de pacientes + ✎ 📄 🗑			
↑	👤	MOVEAT-13	Pesquisar paciente com o nome exato
↑	👤	MOVEAT-14	Pesquisar paciente com nome parcialmente correto
<input type="checkbox"/>	👤	MOVEAT-15	Validar tratamento de erro para pesquisa de paciente com nome inexistente
+ Create quick test			

12. BUGS ENCONTRADOS DURANTE OS TESTES DE SISTEMA

TESTES DE SISTEMA		
ID	DESCRIÇÃO	STATUS
BUG-DASHBOARD-001	Dashboard não exibe nome e e-mail do nutricionista logado (Prioridade Baixa)	Corrigido
BUG-DASHBOARD-002	Nutricionista não é redirecionado após cadastrar novo paciente (Prioridade Baixa)	Corrigido
BUG-DASHBOARD-003	Pesquisa pelo nome exato do paciente não retorna resultado (Prioridade Alta)	Corrigido
BUG-DASHBOARD-004	Pesquisa pelo nome parcial do paciente não retorna resultado (Prioridade Média)	Corrigido

13. MELHORIAS ENCONTRADAS DURANTE OS TESTES DE SISTEMA

TESTES DE SISTEMA		
ID	DESCRIÇÃO	STATUS
UPGRADE-DASH-001	Adicionar "*" nos campos que são obrigatórios (Prioridade Baixa)	Melhoria implementada

III. RELATÓRIO DO TESTE DE REGRESSÃO

Responsável QA: Ana Carollyne Guimarães de Souza

Sistema Operacional: Windows 11 Pro

Ferramenta de teste: Cypress

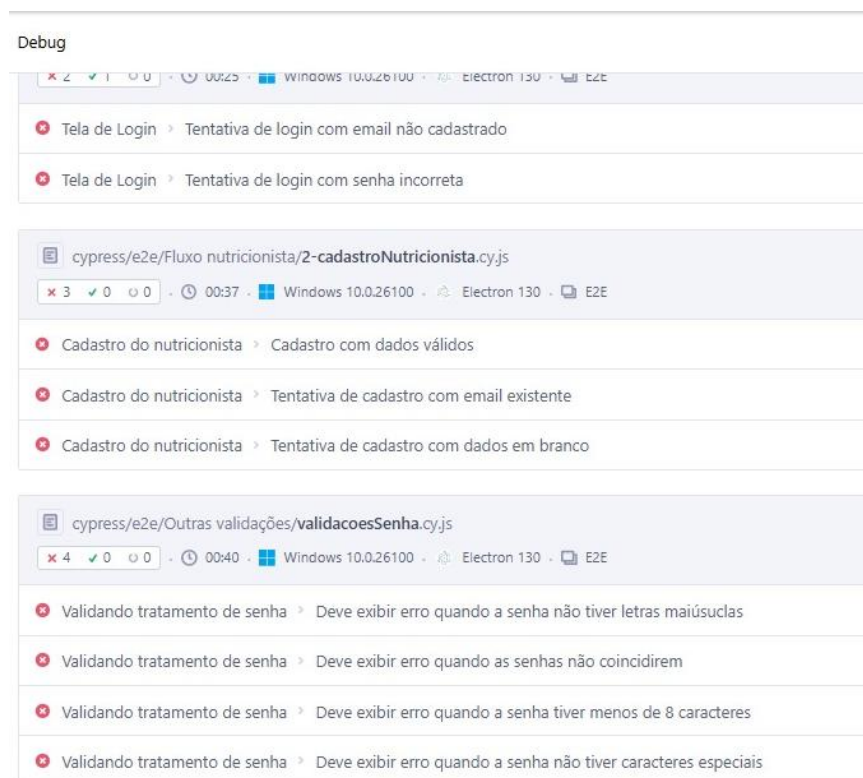
Data dos testes: 06/04/2025

14. OBJETIVO

Validar que o sistema construído anteriormente não tenha “quebrado” após as mudanças realizadas. Todos os casos de teste da sprint de **autenticação** foram automatizados com cypress.

15. RESULTADO DOS TESTES

Após rodar os testes automatizados foram encontrados 10 bugs, sendo 9 deles relacionados aos scripts de teste desatualizados. Estes foram corrigidos e apenas 1, por ser um bug de tela, teve sua issue aberta.



16. BUG ENCONTRADO DURANTE OS TESTES DE REGRESSÃO

TESTES DE REGRESSÃO		
ID	DESCRIÇÃO	STATUS
REGRESSÃO-LOG-001	Cadastro com e-mail e/ou senha inválidos retorna mensagem de erro incorreta (Prioridade Média)	Corrigido