

Autor: Eduardo Costa Miranda Azevedo

Plano de Testes

Objetivo deste :

Baseando-se na norma IEEE Standard for Software Test Documentation, esse plano irá apresentar:

1. Itens a serem testados;
2. **Tipos, técnicas e níveis** de testes a serem realizados para cada um desses itens do item 1;
3. Requerimentos do ambiente para testes (“such as hardware, software, OS, network configurations, tools required.”)

Itens a serem testados

1. Itens da Página Inicial (1.0 - Requisitos lógicos de redirecionamento)

1. a : “A página inicial do sistema, caso o usuário não esteja logado, é a de login”.

Motivo: Pelo planejamento da arquitetura do software, os testes devem certificar que o usuário sem autenticação válida terá acesso somente à tela de login.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência . Os cenários de teste avaliados para esse escopo serão: <ul style="list-style-type: none">• (a) - dado que estou sem autenticação requisito a tela inicial, então devo verificar que é a tela de login• (b) - dado que estou autenticado requisito a tela inicial, então devo verificar que é a tela de listagem de vagas.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

1. b : “Se estiver logado tem que aparecer mensagem de boas vindas, Listagem das vagas”.

Motivo: Pela maneira a qual foram dispostas as telas em congruência com a autenticação do sistema, quando um usuário está logado, ao acessar a página inicial, este deverá observar uma mensagem de boas vindas e a tela inicial, de forma seja fácil de identificar se está autenticado com sucesso e de ter acesso às funcionalidades.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência . Os cenários de teste avaliados para esse escopo serão: <ul style="list-style-type: none">• (a) - dado que insiro credenciais válidas e clico no botão de login, então devo verificar a mensagem de boas vindas e a listagem das vagas.• (b) - dado que estou na tela de descrição de uma vaga e retorno para a página inicial, então devo verificar a mensagem de boas vindas e a listagem de vagas.• (c) - dado que não estou logado e solicito a tela inicial, então devo verificar se fui redirecionado para a tela de login.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

2. Itens do Login

2. a : “Deve ser implementada uma tela de login com campos usuário e senha (ambos obrigatórios)”.

Motivo: Os testes devem certificar que a tela de login contenha os campos usuário e senha e que eles funcionem como o esperado, para que o usuário consiga se autenticar no gerenciador de currículos. Além disso, a funcionalidade de cadastro também deve ser testada, já que ela cria os usuários válidos e aptos a operar o sistema.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
-----------------------	--

Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	<p>A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência. Os cenários de teste avaliados para esse escopo serão:</p> <p style="text-align: center;">Front-End - Login</p> <ul style="list-style-type: none"> • (a) - dado que não preencho o campo email, preencho o campo senha com uma senha válida e clico no botão de login, verifico se aparece uma mensagem de erro e o login não é validado. • (b) - dado que não preencho o campo senha, preencho o campo email com um email válido e clico no botão de login, verifico se aparece uma mensagem de erro e o login não é validado. • (c) - dado que preencho o campo email com um email válido, preencho o campo senha com uma senha válida porém incorreta e clico no botão de login, verifico se aparece uma mensagem de erro e o login não é validado. • (c) - dado que preencho o campo email com um email válido, preencho o campo senha com uma senha válida e correta e clico no botão de login, verifico se aparece a listagem de vagas. • <p style="text-align: center;">Front-End - Registro</p> <ul style="list-style-type: none"> • (a) - dado que preencho o campo email com um email válido, o campo senha com uma senha válida, o campo confirme sua senha com uma senha diferente da primeira senha e clico no botão de registro, verifico se aparece uma mensagem de erro. • (b) - dado que não preencho o campo email, preencho o campo senha com uma senha válida, o campo confirme sua senha com uma senha igual a primeira senha e clico no botão de registro, verifico se aparece uma mensagem de erro. • (c) - dado que preencho o campo email com um email válido, o campo senha com uma senha válida, o campo confirme sua senha com uma senha igual a primeira senha, clico no botão de registrar-me, sou redirecionado para a tela de login depois de um cadastro valido, clico no botão de cadastro, preencho o campo email com o mesmo email utilizado no cadastro passado, preencho a senha com uma senha válida, o campo confirme sua senha com uma senha igual a anteposta, e clico no botão de registrar, então devo verificar que não fui

	<p>redirecionado para nenhuma página e não consegui criar uma nova conta.</p> <ul style="list-style-type: none"> • (d) - dado que preencho o campo email com um email sem domínio, o campo senha com uma senha válida, o campo confirme sua senha com uma senha igual a primeira senha e clico no botão de registro, verifico se aparece uma mensagem de erro. • (e) - dado que preencho o campo email com um email sem pontos, o campo senha com uma senha válida, o campo confirme sua senha com uma senha igual a primeira senha e clico no botão de registro, verifico se aparece uma mensagem de erro. • (f) - dado que preencho o campo email com um email somente com símbolos, o campo senha com uma senha válida, o campo confirme sua senha com uma senha igual a primeira senha e clico no botão de registro, verifico se aparece uma mensagem de erro. • (g) - dado que preencho o campo email com um email válido, não preencho o campo senha e o campo confirme sua senha e clico no botão de registro, verifico se aparece uma mensagem de erro. • (h) - dado que preencho o campo email com um email válido, o campo senha com uma senha válida, o campo confirme sua senha com uma senha igual a primeira senha e clico no botão de registro e em seguida faço o login desse mesmo usuário, verifico se sou direcionado para a tela de listagem de candidatos. <p>Back-End - Login</p> <ul style="list-style-type: none"> • (a) - dado que executo uma requisição com o campo email com um email válido e cadastrado e o campo senha com uma senha válida e correta, verifico se o token de autenticação retornado pela API não é nulo. • (b) - dado que executo uma requisição com o campo email com um email válido porém não cadastrado e o campo senha com uma senha válida, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (c) - dado que executo uma requisição com o campo email com um email válido e cadastrado e o campo senha com uma senha válida porém incorreta, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (d) - dado que executo uma requisição com o campo email com um email em branco e o campo senha com uma senha em branco, verifico se a API de fato
--	--

	<p>não retornou um token mas sim uma lista de erros, e valido esses erros.</p> <ul style="list-style-type: none"> • (e) - dado que executo uma requisição com o campo email com um email nulo e o campo senha com uma senha nula, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. <p style="text-align: center;">Back-End - Registro</p> <ul style="list-style-type: none"> • (a) - dado que crio um usuário, executo uma requisição com o campo email com um email válido, o campo senha com uma senha válida e o campo confirme sua senha com uma senha igual a primeira senha, verifico se o email retornado pela requisição é igual ao enviado e se o ID criado que foi retornado não é nulo. • (b) - dado que executo uma requisição com o campo email com um email nulo, o campo senha com uma senha válida e o campo confirme sua senha com uma senha igual a primeira senha, verifico se foi retornada uma lista de erros pela requisição e valido esses erros. • (c) - dado que executo uma requisição com o campo email com um email válido, o campo senha com uma senha nula e o campo confirme sua senha com uma senha igual a primeira senha, verifico se foi retornada uma lista de erros pela requisição e valido esses erros. • (d) - dado que executo uma requisição com o campo email com um email válido, o campo senha com uma senha em branco e o campo confirme sua senha com uma senha também em branco, verifico se foi retornada uma lista de erros pela requisição e valido esses erros. • (e) - dado que executo uma requisição com o campo email com um email válido, o campo senha com uma senha curta (com menos de 8 dígitos) e o campo confirme sua senha com uma senha igual a primeira senha, verifico se foi retornada uma mensagem de erro pela requisição. • (f) - dado que executo uma requisição com o campo email com um email válido, o campo senha com uma senha longa (com mais de 64 dígitos) e o campo confirme sua senha com uma senha igual a primeira senha, verifico se foi retornada uma lista de erros pela requisição e valido esses erros. • (g) - dado que executo uma requisição com o campo email com um email sem domínio, o campo senha com uma senha válida e o campo confirme sua senha com uma senha igual a primeira senha,
--	--

	<p>verifico se foi retornada uma lista de erros pela requisição e valido esses erros.</p> <ul style="list-style-type: none"> • (h) - dado que executo uma requisição com o campo email com um email em branco, o campo senha com uma senha válida e o campo confirme sua senha com uma senha igual a primeira senha, verifico se foi retornada uma lista de erros pela requisição e valido esses erros. • (i) - dado que executo uma requisição com o campo email com um email já cadastrado, o campo senha com uma senha válida e o campo confirme sua senha com uma senha igual a primeira senha, verifico se foi retornada uma lista de erros pela requisição e valido esses erros.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

2. b : “A tela de login deve ser acessada sempre que um usuário não estiver com uma sessão / token válido e tentar acessar alguma página”.

Motivo: Pelo planejamento da arquitetura do software, os testes devem certificar que ao requisitar-se um acesso à aplicação sem autenticação válida o usuário terá acesso somente à tela de login, para garantir a segurança de acesso.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	<p>A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os títulos de teste avaliados para esse escopo serão:</p> <ul style="list-style-type: none"> • (a) - requisitando a tela de login sem autenticação. • (b) - requisitando a tela cadastro sem autenticação • (c) - requisitando a tela de lista de vagas sem autenticação. • (d) - requisitando a tela de adicionar candidato sem autenticação. • (e) - requisitando PopUp de vincular candidato sem autenticação. • (f) - requisitando a tela de editar candidato sem autenticação. • (g) - requisitando a tela de ver detalhes do candidato

	sem autenticação.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

2. c : “Ao realizar login com sucesso, o usuário deve ser redirecionado para a tela inicial”.

Motivo: Após um login válido, a tela que deve aparecer é a tela inicial, para que o usuário tenha acesso às informações presentes.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os cenários de teste avaliados para esse escopo serão: <ul style="list-style-type: none"> • (a) - dado que preencho o campo email com um email válido e cadastrado, preencho o campo senha com uma senha válida e correspondente ao email cadastrado e clico no botão de login, verifico se sou direcionado para a tela de listagem de candidatos(tela inicial) e válido a mensagem de Listagem de vagas.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

3. Itens da Listagem de Vagas

3. a : “Devem ser listados vagas com os dados que vem da API do Compleo”.

Motivo: Os testes devem garantir que os dados das vagas listadas sejam provenientes da API do Compleo, assegurando que os dados estão sendo passados corretamente.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos

Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os cenários de teste avaliados para esse escopo serão: <ul style="list-style-type: none"> • (a) - dado que estou autenticado e clico no botão candidatos e depois no botão dashboard, verifico se sou direcionado para a tela de listagem de candidatos.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

3. b : “Deve ter um botão de vincular Candidato que leva para um popup que abre a listagem de candidatos e um botão de vincular”.

Motivo: Para que o usuário possa vincular candidatos, deve haver um botão que abrirá a lista de candidatos em um popup, cada candidato com seu botão de vincular.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os cenários de teste avaliados para esse escopo serão: <ul style="list-style-type: none"> • (a) - dado que o usuário já está autenticado e se encontra na tela de listagem de vagas(tela inicial da aplicação), clico no botão vincular candidatos e depois no botão vincular e verifico se a mensagem do botão mudou de vincular para desvincular (o que mostra que o candidato foi vinculado). • (a) - dado que o usuário já está autenticado e se encontra na tela de listagem de vagas(tela inicial da aplicação), clico no botão vincular candidatos e depois no botão vincular e verifico se a mensagem do botão mudou de vincular para desvincular (o que mostra que o candidato foi vinculado), feito isso clico no botão desvincular desse mesmo candidato e vejo se a mensagem do botão mudou para vincular, o que mostra que consegui desvincular.
Nível de teste	O nível de teste é de unidade e deve ocorrer após o

(quando)	desenvolvimento desta funcionalidade e após cada alteração no software.
----------	---

4. Itens da Listagem de Currículos

4. a : “Deve ter uma lista de todos os candidatos cadastrados no sistema com os seguintes dados:

- i. Nome
- ii. Cargo
- iii. Data de nascimento
- iv. Senioridade

”.

Motivo: Os candidatos cadastrados devem conter nome, cargo, data de nascimento, e senioridade aparecendo na lista, para que o usuário consiga visualizá-los apropriadamente.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os títulos de teste que passarão por esse escopo serão: Back-End - Listagem de vagas <ul style="list-style-type: none"> • (a) - requisitando a listagem de vagas com autenticação. • (b) - requisitando a listagem de vagas sem autenticação. Front-End - Listagem de vagas <ul style="list-style-type: none"> • (a) - adicionar candidato. • (b) - editar candidato. • (c) - vincular candidato. • (d) - desvincular candidato.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

4. b : “Quando clicar no candidato tem que abrir o detalhamento (item 5)”.

Motivo: Para que o usuário tenha acesso à mais informações, ao clicar no candidato, o detalhamento sobre ele deve abrir.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional : visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os títulos de teste que envolvem esse escopo serão: <ul style="list-style-type: none"> • (a) - Adicionar candidato válido. • (b) - Editar candidato válido.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

4. c : “Deve possuir um botão de adicionar que leva para a tela de cadastro(item 6)”.

Motivo: Para que seja possível cadastrar um novo candidato, deve haver um botão adicionar que leva para a tela de cadastro.

Essa funcionalidade será testada juntamente aos cenários e títulos do item (6).

4. d : “Deve ter do lado da listagem um botão de editar que leva a mesma tela de adicionar somente com os campos já populados”.

Motivo: Na lista de candidatos, deve haver um botão de editar para cada candidato, que levará para a mesma tela de cadastrar com os campos já preenchidos, para que o usuário possa editá-lo.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional: visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os cenários de teste avaliados para esse escopo serão: <ul style="list-style-type: none">• (a) - dado que o usuário já foi logado e um candidato adicionado, procuro-o na lista de candidatos, cliço no botão editar e edito apenas o telefone. Então procuro novamente o candidato pelo nome e verifico se os nomes continuam iguais e os telefones mudaram.• (b) - dado que o usuário já foi logado e um candidato adicionado, procuro-o na lista de candidatos, cliço no botão editar e edito apenas o nome. Então procuro novamente o candidato pelo novo nome e verifico se os telefones continuam iguais e os nomes mudaram.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

5. Itens do Detalhe dos Candidatos

5. a : “Criar página com os seguintes campos:

- i. Nome
- ii. Dados pessoais (CPF, Data nascimento, Endereço, etc) conforme currículo
- iii. Dados Escolares (conforme currículo)
- iv. Experiências (Conforme currículo)
- v. Link para download do currículo em arquivo
- vi. Senioridade
- vii. Cargo

”.

Motivo: Deve haver uma página de detalhamento do currículo para cada

candidato, de forma que o usuário consiga visualizar todos os dados: Nome, Dados Pessoais, Dados Escolares, Experiências, Link para download do currículo em arquivo, Senioridade e Cargo.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional : visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os cenários de teste avaliados para esse escopo serão: Front-End <ul style="list-style-type: none">• (a) - Dado que estou na tela de candidatos e clico no nome de algum candidato visível, sou redirecionado para a tela de detalhes e devo verificar os campos evidentes. Ademais, essa tela é utilizada para auxiliar na validação dos cenários de adição de candidato, edição de candidato, deleção de candidato, listagem de candidato e vinculação/desvinculação de candidato à determinada vaga.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.

6. Itens do Cadastro dos Candidatos

6. a : “Devem ser aberto um formulário com os seguintes campos:

- i. Nome
- ii. Dados pessoais (CPF, Data nascimento, Endereço, etc) conforme currículo
- iii. Dados Escolares (conforme currículo)
- iv. Experiências (Conforme currículo)
- v. Campo para upload de arquivos
- vi. Senioridade
- vii. Cargo
- viii. (<https://www.callicoder.com/spring-boot-file-upload-download-rest-api-example/>)

”

Motivo: Para que seja possível cadastrar os candidatos, deve haver uma tela de cadastro que contém os campos: Nome, Dados Pessoais, Dados Escolares, Experiências, Campo para upload do currículo em arquivo,

Senioridade e Cargo.

Tipo de teste (o que)	De funcionalidade (prioriza navegações e interações e checa se a regra de negócio está sendo respeitada; códigos que podem realizar buscas no banco de dados e devem realizar verificações)
Técnica de teste (como)	Funcional : visa analisar as funcionalidades da aplicação e como ela se comporta diante de cenários adversos
Critério (a partir da técnica)	<p>A partir da técnica funcional, o critério é o teste baseado nos valores limites otimizado pelo princípio das classes de equivalência os cenários de teste avaliados para esse escopo serão:</p> <p style="text-align: center;">Front-End</p> <ul style="list-style-type: none"> • (a) - dado que preencho todos os campos de dados pessoais com dados válidos (Nome, Data de Nascimento, Telefone, Cargo, Senioridade) exceto CPF, que é deixado em branco, e clico no botão próximo, verifico se aparece uma mensagem de erro. • (b) - dado que preencho todos os campos de dados pessoais com dados válidos (Nome, Data de Nascimento, Telefone, Cargo, Senioridade) exceto CPF, que é preenchido com um CPF curto (com menos de 11 dígitos), e clico no botão próximo, verifico se aparece uma mensagem de erro. • (c) - dado que preencho todos os campos de dados pessoais com dados válidos (Nome, CPF, Telefone, Cargo, Senioridade) exceto Data de Nascimento, que é deixado em branco, e clico no botão próximo, verifico se aparece uma mensagem de erro. • (d) - dado que preencho todos os campos de dados pessoais com dados válidos (Nome, Data de Nascimento, CPF, Cargo, Senioridade) exceto Telefone, que é deixado em branco, e clico no botão próximo, verifico se aparece uma mensagem de erro. • (e) - dado que preencho todos os campos de dados pessoais com dados válidos (Nome, Data de Nascimento, CPF, Cargo, Senioridade) exceto Telefone, que é preenchido com um telefone curto (com menos de 11 dígitos), e clico no botão próximo, verifico se aparece uma mensagem de erro. • (f) - dado que o usuário já foi logado, clico no botão candidato e depois no botão adicionar e preencho todos os campos com dados válidos e então verifico se o nome do candidato adicionado está na lista de candidatos. <p style="text-align: center;">Back-End</p>

	<ul style="list-style-type: none"> • (a) - dado que executo uma requisição com o campo telefone com um telefone longo (com mais de 11 dígitos), verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (b) - dado que executo uma requisição com o campo telefone com um telefone curto (com menos de 11 dígitos), verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (c) - dado que executo uma requisição com o campo telefone com um telefone em branco, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (d) - dado que executo uma requisição com o campo telefone com um telefone nulo, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (e) - dado que executo uma requisição com o campo Data de Nascimento com um data após o dia atual, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (f) - dado que executo uma requisição com o campo Data de Nascimento com um data em branco, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (g) - dado que executo uma requisição com o campo Data de Nascimento com um data nula, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (h) - dado que executo uma requisição com o campo CPF com um CPF longo (com mais de 11 dígitos), verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (i) - dado que executo uma requisição com o campo CPF com um CPF curto (com menos de 11 dígitos), verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (j) - dado que executo uma requisição com o campo CPF com um CPF em branco, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (k) - dado que executo uma requisição com o campo CPF com um CPF nulo, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (l) - dado que executo uma requisição com o campo CEP com um CEP longo (com mais de 8 dígitos), verifico se a API de fato não retornou um token mas
--	--

	<p>sim uma lista de erros, e valido esses erros.</p> <ul style="list-style-type: none"> • (m) - dado que executo uma requisição com o campo CEP com um CEP curto (com menos de 8 dígitos), verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (n) - dado que executo uma requisição com o campo CEP com um CEP em branco, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (o) - dado que executo uma requisição com o campo CEP com um CEP nulo, verifico se a API de fato não retornou um token mas sim uma lista de erros, e valido esses erros. • (p) - dado que executo uma requisição com os campos populados corretamente e respeitando os requisitos impostos pela regra de negócio verifico o objeto retornado e se os campos oriundos da requisição conferem com os campos enviados.
Nível de teste (quando)	O nível de teste é de unidade e deve ocorrer após o desenvolvimento desta funcionalidade e após cada alteração no software.