

Plano de Testes
Desenvolvimento de Sistemas

Professor:

Tell Marcus de Souza Moitas

Equipe:

Alanis Isabelle de O. Silva

Filipe de Oliveira Ataíde

Victor Augusto Fernandes Pereira

Thales Azevedo Silva

Junho / 2021

Histórico de Revisões

Data	Versão	Descrição	Autor
10/julho/2021	1.0	Release Inicial	Alanis I. O. S., Filipe de O.A., Thales A.. S., Victor A. F. P.
11/julho/2021	1.0.1	Continuação da Release inicial (edição do item 1)	Alanis I. O. S., Filipe de O.A., Thales A.. S., Victor A. F. P.
16/julho/2021	1.0.2	Ajuste do item 1	Alanis I. O. S., Filipe de O.A., Thales A.. S., Victor A. F. P.
28/julho/2021	1.0.3	Revisão do item 2	Alanis I. O. S., Filipe de O.A., Thales A.. S., Victor A. F. P.

Índice

1. INTRODUÇÃO	4
1.1 OBJETIVOS	4
1.2 O COOKIT	4
1.3 ESCOPO	4
1.4 IDENTIFICAÇÃO DE PROJETO	5
2. REQUISITOS A TESTAR	5
2.1 Tipos de Teste	5
2.1.1 TESTE DE CAIXA PRETA: BUSCA POR FILTRO AVANÇADO	5
2.1.2 Teste de Caixa preta: Crude receita + permissão de usuário	6
2.1.3 Teste de caixa preta: Notificações	7
2.1.4 Teste de caixa preta: Login	7
2.1.5 Teste de caixa preta: Cadastro	7
2.1.6 Teste de Caixa preta: Autenticação	8
2.1.7 Teste de Caixa Preta: Cadastro de Receita	8
2.1.8 Teste de Caixa Preta: Comentário	9
2.1.9 Teste de Caixa Preta: Busca simples	9
3. ESTRATÉGIA DE TESTE	0
3.1 TIPOS DE TESTE	0
3.1.1 <i>Teste de Integridade de Dados e do Banco de Dados</i>	0
3.1.2 <i>Teste de Função</i>	0
3.1.3 <i>Teste da Interface do Usuário</i>	0
3.1.4 <i>Teste de Performance</i>	10
3.1.5 <i>Teste de Carga</i>	11
3.1.6 <i>Teste de Segurança e Controle de Acesso</i>	11
3.1.7 <i>Teste de Instalação</i>	12
3.2 FERRAMENTAS	12
4. RECURSOS	13
4.1 TRABALHADORES	13

4.2 SISTEMA	14
5. CRONOGRAMA	15

1. Introdução

1.1 Objetivos

Esse documento do Plano de Testes do Cook It. compõe-se dos seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os Requisitos a Testar recomendados (alto nível).
- 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.

1.2 O Cookit

O software **Cook It** é um sistema web pensando em facilitar o preparo de alimentos mediante a necessidade do usuário que não encontra praticidade durante o processo. A aplicação remove barreiras que dificultam ao fazer pratos que não envolvam os ingredientes que possui em mãos sugerindo receitas.. CookIt tem por sua vez o intuito de proporcionar ao usuário o acesso facilitado através de filtros de busca que o estimulem a cozinhar receitas tirando a necessidade de comer fora de casa.

1.3 Escopo

1. O teste de instalação e configuração não serão aplicados ao projeto, pois trata-se de uma aplicação web, não necessitando de instalação.
2. O teste de Integridade será incluso visando testar o que ocorrerá com o sistema quando houver muitas receitas e ingredientes.
3. O Teste de segurança ocorrerá tendo em vista checar se as restrições de acesso está sendo respeitadas com base na hierarquia de usuário.
4. O funcional checar se o sistema está de acordo com a documentação de casos de uso. V

5. Ocorreram testes de unidade a fim de verificar se os trechos isolados do projeto. Em sequência será realizado testes de integração, a fim de observar se mais de um componente funcionam juntos.
6. Entre os testes de performance ocorreram o de carga, estresse e estabilidade
 - a. 4, 10, 100, 500 usuários simultâneos conseguem fazer login, upload de uma receita, visualizando uma receita, em menos de 1 segundo.
7. Também ocorrerá o teste de usabilidade, a fim de validar o layout idealizado para o sistema. Ocorrerá também o teste de regressão a visando a prevenção do re-surgimento de erros.V

Testes críticos

- Performance
- Segurança
- Integridade

1.1 Identificação de Projeto

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de testes:

Documento	Criado ou Disponível	Recebido ou Revisado
Especificação de Requisitos?	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Documento de Visão	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Casos de Uso	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Projeto?	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Documento de Arquitetura	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Protótipo	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Manual do Usuário	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não

Documento de Riscos	■ Sim □ Não	■ Sim □ Não
Glossário	■ Sim □ Não	■ Sim □ Não

2. Requisitos a Testar

2.1 Tipos de Teste

Nota: As transações abaixo se referem às “transações lógicas de negócio”. Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma dada informação.

2.1.1 Teste de Caixa preta: Busca por filtro avançada

Autor: Alanis Isabelle de Oliveira Silva

ENTRADAS	SAÍDAS
Ingrediente X colocado no campo “quero que tenha”	Serão exibidas as receitas com ingredientes X;
Ingrediente Y colocado no campo “quero que tenha”	Serão exibidas as receitas com ingredientes Y;
Ingrediente Y colocado no campo “quero que <u>não</u> tenha”	Serão exibidas as receitas que não contém Y
Ingrediente X e Y colocados no campo “quero que tenha”	Serão exibidas as receitas com ingredientes X e Y;
Ingrediente Y colocado no campo “quero que tenha” e “quero que <u>não</u> tenha”	O Sistema deve informar a mensagem: não se pode colocar Y em ambos os campos;

Ingrediente X colocado no campo “ quero que tenha ” e Y colocado no campo “ quero que <u>não</u> tenha ”	O Sistema exibirá receitas com X que não contenham Y.
Selecionada a dificuldade F	O Sistema exibirá apenas receitas com a dificuldade F;
Ingrediente X colocado no campo “ quero que tenha ” e selecionada a dificuldade F	Serão exibidas as receitas com ingredientes X e com a dificuldade F;
Ingrediente Y colocado no campo “ quero que <u>não</u> tenha ” Selecionada a dificuldade F	Serão exibidas as receitas que não contenham Y e com a dificuldade F;
Ingrediente X colocado no campo “ quero que tenha ”, Ingrediente Y colocado no campo “ quero que <u>não</u> tenha ” e selecionada a dificuldade F	Serão exibidas as receitas que contenham X, não contenham Y e com a dificuldade F;
Selecionado o Sabor Doce	Serão exibidas todas as receitas do tipo doce;
Selecionado o Sabor Salgado	Serão exibidas todas as receitas do tipo salgada;
Selecionado a Categoria B	Serão exibidas todas as receitas da categoria B;
Selecionado a Categoria A e B	Serão exibidas todas as receitas das categorias A e B;

2.1.2 Teste de Caixa preta: Crude receita + permissão de usuário

Autor: Thales Azevedo Silva

O perfil deverá ter permissão de efetuar CRUD apenas de suas receitas.

ENTRADAS	Saídas
O usuário clica em adicionar receita	A receita é adicionada ao banco e vinculada ao perfil.
O usuário tenta alterar uma receita que não foi cadastrada por ele.	Não é possível alterar uma receita que não esteja vinculada ao perfil do usuário em questão.

O usuário tenta alterar uma receita vinculada ao seu perfil.	O sistema permite a edição e salva as alterações desejadas.
O usuário tenta excluir uma receita não vinculada ao seu perfil	O sistema não possibilita a exclusão de uma receita que não esteja vinculada ao usuário.
O usuário tenta excluir uma receita vinculada ao seu perfil.	A exclusão é efetuada com sucesso e a lista de receitas é atualizada.

2.1.3 Teste de caixa preta: Notificações

Autor: Thales Azevedo Silva

O sistema deverá notificar ações feitas por outros usuários ou do próprio sistema e por consequência o usuário irá visualizá-las.

ENTRADAS	Saídas
O usuário clica no ícone de notificação	O sistema retorna uma listagem com todas as notificações
O usuário clica em uma notificação da listagem	O sistema retorna o detalhe da notificação selecionada
O usuário visualiza o ícone de notificações com a quantidade atualizada e clica para ver a listagem.	O sistema retorna a listagem separando a lista de notificações lidas e não lidas pelo usuário.
O usuário tenta visualizar os detalhes de uma notificação anteriormente clicada.	O sistema retorna a listagem de notificações lidas e mostra o botão de detalhes desabilitado, não permitindo visualização do usuário novamente.

2.1.4 Teste de caixa preta: Login

Autor: Filipe de Oliveira Ataíde

Verificar se o login do sistema está funcionando adequadamente, tanto no Django Framework

como no Vue JS.

Login:

Cenários	1	2	3
Usuário Consta no Sistema	Sim	Não	Sim
Senha Consta no Sistema	Sim	Não	Não
Saída Esperada	Logado com Sucesso	Usuário não existe	Senha Incorreta

2.1.5 Teste de caixa preta: Cadastro

Autor: Filipe de Oliveira Ataíde

Verificar se o cadastro do sistema está funcionando adequadamente, tanto no Django Framework como no Vue JS.

Entrada	Saída
Campos estão preenchidos adequadamente	Conta criada com sucesso.
Campos obrigatórios não estão todos preenchidos	Campos obrigatórios são marcados em vermelho para que o usuário os preencha
Senha não atende aos requisitos	O campo de senha é marcado em vermelho com um pequeno texto informando os requisitos
Nome de usuário já existe	O sistema informará que um nome de usuário diferente deverá ser informado um para poder cadastrar a conta.
E-mail já está cadastrado	O sistema informará que o e-mail já está

	cadastrado no sistema
--	-----------------------

2.1.6 Teste de Caixa preta: Autenticação

Autor: Filipe de Oliveira Ataíde

Procurar pelas configurações da conta (dados do perfil, desativação de conta), e tentar o mesmo com o perfil de terceiros cadastrados no site a fim de descobrir se o acesso é permitido para o usuário logado e bloqueado para o resto. Isso deverá ser testado tanto por interface como por URL.

Cenários	Caso 1	Caso 2	Caso 3
Usuário está autenticado	Sim	Sim	Não
Usuário a ser editado é o autenticado	Sim	Não	Não
Saída	Página de Configurações é exibida	Página de Configurações do usuário autenticado é exibida	Página não é exibida

2.1.7 Teste de Caixa Preta: Cadastro de Receita

Autor: Victor Augusto Fernandes Pereira

Primeiramente, o teste se inicia com o cadastro de uma receita no sistema, onde logo após isso se deve exibir a receita por completo para averiguar se tudo que foi cadastrado está corretamente lá.

Cenário	Saída
Campos preenchidos corretamente	Receita cadastrada com sucesso
Campos preenchidos corretamente	Receita cadastrada com erros
Campos faltando serem preenchimento da receita	Os campos que estiverem faltando para serem preenchidos na receita ficaram vermelhos
Nenhum campo preenchido na receita	Todos os campos ficam em vermelho

2.1.8 Teste de Caixa Preta: Comentário na Receita

Autor: Victor Augusto Fernandes Pereira

Primeiramente, o teste se inicia com o usuário comentando em uma receita logo após averiguando se a mesma foi salva pelo sistema na parte dos comentários.

Cenário	Saída
Campo do comentário da receita preenchido.	Comentário cadastrado na receita com sucesso.
Nenhum comentário preenchido no campo dos comentários.	Sem comentários do usuário na receita

2.1.9 Teste de Caixa Preta: Busca Simples

Autor: Alanis Isabelle de Oliveira Silva

O teste consiste em buscar parte de um nome de uma receita e o sistema retornar todas as receitas como aquela palavra.

ENTRADAS	Saídas
X	O sistema lista todas as receitas que possuem a palavra “X”

2.1.10 Teste de Caixa Preta: Teste de integração Login + Cadastro

Autor: Alanis Isabelle de Oliveira Silva

O teste consiste em testar as funcionalidades Login e Cadastro simultaneamente e ver se o sistema lida bem com a concorrência de ações.

ENTRADAS	Saídas
Login de conta A + Cadastro de conta B	O sistema consegue executar o login de conta A e Cadastro de conta B

3. ESTRATÉGIA DE TESTE

4. Tipos de Teste

Nota: As transações abaixo se referem às “transações lógicas de negócio”. Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma dada informação.

4.1 *Teste de Integridade de Dados e do Banco de Dados*

Objetivo do Teste:	Garantir que os métodos e processos de acesso ao banco de dados funcionam apropriadamente e sem corrupção dos dados.
Técnica:	<ul style="list-style-type: none">▪ Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos.▪ Inspecionar o banco de dados para garantir que os dados foram populados como pretendido, que todos os eventos do banco de dados ocorreram apropriadamente, ou revisar os dados retornados para garantir que os dados corretos foram recuperados pelas razões corretas.
Critério de Finalização:	Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados.
Considerações Especiais:	<ul style="list-style-type: none">▪ Nenhum

1.1.1 Teste de Função

Objetivo do Teste:	Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento, e recuperação.
Técnica:	<p>Executar cada caso de uso, fluxo de caso de uso, usando dados válidos e inválidos, para verificar o seguinte:</p> <ul style="list-style-type: none">▪ Os resultados esperados ocorrem quando dados válidos são usados▪ As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados.▪ Cada regra de negócio é aplicada apropriadamente
Critério de Finalização:	<ul style="list-style-type: none">▪ Todos os testes planejados foram executados.▪ Todos os defeitos identificados foram tratados.
Considerações Especiais:	Nenhum

4.2 Teste da Interface do Usuário

Objetivo do Teste:	<p>Verificar o seguinte:</p> <ul style="list-style-type: none">▪ A navegação através dos alvos de teste reflete as funções e os requisitos do negócio apropriadamente, incluindo janela-a-janela, campo-a-campo, e o uso de métodos de acesso (tecla tab, movimentos do mouse, teclas aceleradoras)▪ Objetos e características da janela, tais como menus, tamanho, posição, estado e foco conformam-se aos padrões.
--------------------	---

Técnica:	Criar ou modificar os testes para cada janela para verificar a navegação e os estados de objeto apropriados para cada janela e objetos da aplicação.
Critério de Finalização:	É verificado que cada janela permanece consistente com a versão de comparação ou dentro de padrões aceitáveis.
Considerações Especiais:	Nem todas as propriedades para objetos personalizados e terceirizados podem ser acessadas.

4.3 *Teste de Performance*

Objetivo do Teste:	Verificar que os comportamentos de performance para as transações designadas ou funções de negócio sob as seguintes condições: <ul style="list-style-type: none"> ▪ Carga de trabalho normal antecipada ▪ Carga de trabalho no pior caso antecipada
Técnica:	<ul style="list-style-type: none"> ▪ Utilizar o módulo Selenium com o Firefox/Chrome.
Critério de Finalização:	Finalização bem sucedida dos scripts de teste sem qualquer falha e dentro da alocação de tempo aceitável.
Considerações Especiais:	Nenhum

4.4 *Teste de Carga*

Objetivo do Teste:	Verifique o tempo de resposta para as transações designadas ou casos de negócio sob condições variantes de carga de trabalho.
Técnica:	<ul style="list-style-type: none"> ▪ Use testes desenvolvidos para o Teste do Ciclo de Negócio ou Função.

	<ul style="list-style-type: none"> ▪ Modifique os arquivos de dados para aumentar o número de transações ou os testes para aumentar o número de vezes que cada transação ocorre.
Critério de Finalização:	Múltiplas transações ou usuários: finalização bem sucedida dos testes sem qualquer falha e dentro da alocação de tempo aceitável.
Considerações Especiais:	<ul style="list-style-type: none"> ▪ O teste de carga deve ser executado em uma máquina dedicada ou em um tempo dedicado. Isso permite o controle total a e mensuração precisa. ▪ As bases de dados usadas para os testes de carga devem ou ser do tamanho real ou igualmente dimensionadas.

4.5 *Teste de Segurança e Controle de Acesso*

Objetivo do Teste:	<ul style="list-style-type: none">● Segurança do Nível de Aplicação: Verifique que um ator pode acessar apenas aquelas funções ou dados para os quais o seu tipo de usuário tem permissão.● Segurança do Nível de Sistema: Verifique que apenas aqueles atores com acesso ao sistema e aplicações têm permissão de acessá-los.
Técnica:	<ul style="list-style-type: none">● Segurança do Nível de Aplicação: Identifique e liste cada tipo de usuário e as funções ou dados para os quais cada tipo tem permissão.● Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicas para cada tipo de usuário.● Modifique o tipo de usuário e repita os testes para os mesmos usuários. Em cada caso, verifique que funções ou dados adicionais estão corretamente disponíveis ou negados.● Acesso de Nível de Sistema: Ver Considerações Especiais abaixo.
Critério de Finalização:	Para cada tipo de ator conhecido as funções ou dados apropriados estão disponíveis, e todas as transações funcionam como esperado e rodam nos Testes de Função anteriores.
Considerações Especiais:	O Acesso ao sistema deve ser revisado ou discutido com o administrador de rede ou de sistema apropriado. Esse teste pode não ser necessário já que ele pode ser uma função da administração da rede ou sistema.

4.6 *Teste de Instalação*

Objetivo do Teste:	<p>Verifique que os alvos de teste instalam apropriadamente em cada configuração de hardware necessária sobre as seguintes condições:</p> <ul style="list-style-type: none">• Uma nova instalação, em um nova máquina, que nunca fora anteriormente instalada com o TIG.• atualização, numa máquina onde o TIG já fora previamente instalado, para a mesma versão• atualização, numa máquina que já disponha do TIG instalado, de uma versão mais velha
Técnica:	<p>Manualmente desenvolve scripts automatizados, para validar a condição da máquina alvo – novo TIG nunca instalado; TIG na mesma versão ou versão mais velha já instalada.</p> <p>Começar ou executar a instalação</p> <p>Usando um subgrupo predeterminado de scripts de teste de funções, rode as transações.</p>
Critério de Finalização:	<p>As transações do TIG executam de forma bem sucedida, sem falha.</p>
Considerações Especiais:	<p>Quais transações do TIG devem ser seleccionadas para abranger um teste de confiança de que a aplicação TIG foi instalada de forma bem sucedida e que nenhum componente importante de software está faltando?</p>

5. Ferramentas

As seguintes ferramentas serão empregadas para esse projeto:

	Ferramenta	Vendedor
Gerenciamento de Teste	Django Test, Selenium	Python, Selenium
Projeto de Teste		
Gerenciamento de Projeto	Google Docs	Google
Ferramentas do SGBD	pgAdmin	pgAdmin

6. Recursos

Essa seção apresenta os recursos recomendados para o projeto do CookIt, suas principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

6.1 Trabalhadores

Essa tabela mostra as suposições de recrutamento para o projeto.

Recursos Humanos		
Trabalhador	Recursos Mínimos Recomendados	Responsabilidades Específicas ou Comentários
Gerente de Teste, Gerente do Projeto de Teste	Alanis Isabelle Filipe Oliveira Thales Azevedo Victor Augusto	Fornece supervisionamento gerencial. Responsabilidades: <ul style="list-style-type: none">• provê direcionamento técnico• adquire recursos apropriados• fornece relatórios de gerenciamento
Test Designer	Alanis Isabelle Filipe Oliveira Thales Azevedo Victor Augusto	Identifica, prioriza, e implementa os casos de teste. Responsabilidades: <ul style="list-style-type: none">• gera o plano de teste• cria o modelo de teste• avalia a efetividade do esforço de teste
Testador	Alanis Isabelle Filipe Oliveira Thales Azevedo Victor Augusto	Executa os testes. Responsabilidades: <ul style="list-style-type: none">• executar os testes• registrar os resultados• restabelecer-se dos erros

		<ul style="list-style-type: none"> documentar solicitações de mudança
Administrador do Sistema de Teste	Alanis Isabelle Filipe Oliveira Thales Azevedo Victor Augusto	Garante que o ambiente e os bens de teste sejam gerenciados e mantidos. Responsabilidades: <ul style="list-style-type: none"> administrar o sistema de gerenciamento teste instalar e gerenciar o acesso do trabalhador ao sistema de testes
Gerente do Banco de Dados, Administrador do Banco de Dados	Alanis Isabelle Filipe Oliveira Thales Azevedo Victor Augusto	Garante que o ambiente e bens de teste de dados (banco de dados) sejam gerenciados e mantidos. Responsabilidades: <ul style="list-style-type: none"> administrar os dados de teste (base de dados)
Designer	Alanis Isabelle Filipe Oliveira Thales Azevedo Victor Augusto	Identifica e define as operações, atributos, e associações das classes de teste. Responsabilidades: <ul style="list-style-type: none"> identificar e definir as classes de teste identificar e definir os pacotes de teste
Implementador	Alanis Isabelle Filipe Oliveira Thales Azevedo Victor Augusto	Implementa e faz os testes unitários das classes e pacotes de teste. Responsabilidades:

		<ul style="list-style-type: none"> cria as classes e pacotes de teste implementados no modelo de teste
--	--	---

6.2 Sistema

A tabela seguinte expõe os recursos do sistema para o projeto de teste.

Recursos do Sistema
Servidor de Banco de Dados Postgres PgAdmin
Terminais Clientes Azure
Repositório de Testes 1 PC 3 PCs de Desenvolvimento de Teste

7. Cronograma

Milestone	Data de Início	Data de Término
Planejar Teste	10/06/21	03/07/21
Projetar Teste	17/06/21	03/07/21
Implementar Teste	08/07/21	28/07/21
Executar Teste	08/07/21	28/07/21
Avaliar Teste	28/07/21	30/07/21

ANEXO I - Capturas de tela da execução dos testes

Teste Unitário 01:

Título: Teste unitário CRUD receita por sessão (test.py)

Ferramenta: Django

Link:

<https://gitlab.devops.ifrn.edu.br/tads.cnat/pdscorporativo/2021.1/cookit/cookit-api/-/blob/usuario+api/cookit/receita/tests.py>

Autor: Thales Azevedo Silva

Data da execução: 28/07/2021

```
thales@thales-PC:~/Música/project/Teste_Software/cookit-api/cookit$ python3 manage.py test
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
....
-----
Ran 4 tests in 0.011s

OK
Destroying test database for alias 'default'...
```

Teste Unitário 02:

Título: Teste unitário Cadastro de Usuário

Ferramenta: Unittest/Django Test

Autor: Filipe de Oliveira Ataíde

Data da execução: 28/07/2021

```
python manage.py test usuario.tests.UnitarioTestCase.test_unitario
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 0.004s

OK
Destroying test database for alias 'default'...
```

Teste Unitário 03:

Título: Teste unitário Cadastro de Receita(test.py)

Ferramenta: Unittest/Django Test

Autor: Victor Augusto Fernandes Pereira

Link: [receita/tests.py · usuario+api · tads.cnat / pdscorporativo / 2021.1 / Cookit / cookit-api · GitLab \(ifrn.edu.br\)](#)

Data da execução: 28/07/2021

```
(project-name) λ python manage.py test receita
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
-----
Ran 3 tests in 0.599s

OK
Destroying test database for alias 'default'...
```

Testes de Sistema

Teste Sistema 01:

Título: CRUD receita por sessão (test.py)

Ferramenta: Selenium

Link:

<https://gitlab.devops.ifrn.edu.br/tads.cnat/pdscorporativo/2021.1/cookit/cookit-api/-/blob/usuario+api/cookit/receita/tests.py>

Autor: Thales Azevedo Silva

Data da execução: 28/07/2021

```
# Teste de sistemas CRUD por sessão - Thales Azevedo Silva

class create_user(TestCase):
    def test_login_session(self):
        self.driver = Firefox()
        self.driver.get('http://localhost:8080/Login')
        self.driver.find_element_by_name('email').send_keys("luan@gmail.com")
        self.driver.find_element_by_name('password').send_keys("meteorodapaixao")
        self.driver.find_element_by_name('submit_form').click()

    def test_cadastro_session(self):
        self.driver = Firefox()
        self.driver.get('http://localhost:8080/FormReceita/')
        self.driver.find_element_by_name('nome_receita').send_keys("ovo de pascoa")
        self.driver.find_element_by_name('categoria').send_keys("L")
        self.driver.find_element_by_name('sabor_receita').send_keys("D")
        self.driver.find_element_by_name('dificuldade').send_keys("M")
        self.driver.find_element_by_name('porcoes').send_keys(2)
        self.driver.find_element_by_name('tempo_preparo').send_keys(50)
        self.driver.find_element_by_name('tempo_unidade_medida').send_keys("M")
        self.driver.find_element_by_name('ingrediente').send_keys("chocolate")
        self.driver.find_element_by_name('add-button').click()
        self.driver.find_element_by_name('quantidade_ingrediente').send_keys(5)
        self.driver.find_element_by_name('unidade_medida_ingrediente').send_keys("U")
        self.driver.find_element_by_name('modo_preparo').send_keys("Esquente o chocolate")
        self.driver.find_element_by_name('observacoes_adicionais').send_keys("comentário")
        self.driver.find_element_by_name('submit-button').click()
```

Teste Sistema 02:

Título: CRUD receita por sessão (test.py)

Ferramenta: Selenium

Link:

<https://gitlab.devops.ifrn.edu.br/tads.cnat/pdscorporativo/2021.1/cookit/cookit-api/-/blob/usuario+api/cookit/receita/tests.py>

Autor: Thales Azevedo Silva

Data da execução: 28/07/2021

```

def test_list_session(self):
    self.driver = Firefox()
    self.driver.get('http://localhost:8080/')
    self.driver.find_element_by_name('button-detail').click()

def test_update_session(self):
    self.driver = Firefox()
    self.driver.get("http://localhost:8080/EditarReceita/1")
    self.driver.find_element_by_name('nome_receita').send_keys("ovo de pascoa trunfado")
    self.driver.find_element_by_name('submit-button').click()

```

Teste Sistema 03:

Título: Login (login.py)

Ferramenta: Selenium

Autor: Filipe de Oliveira Ataíde

Data da execução: 28/07/2021

```

(venv) filipe@filipe-Aspire-A315-53:~/Documentos/PDS Corporativo/cookit-api/cookit$
python manage.py test usuario.tests.LoginTestCase.test_login
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 15.054s

OK
Destroying test database for alias 'default'...

geckodriver.log x
1627690334122 geckodriver INFO Listening on 127.0.0.1:59803
1627690334128 mozrunner:runner INFO Running command: "/usr/bin/firefox" "--
marionette" "--foreground" "--no-remote" "--profile" "/tmp/rust_mozprofileqa880X"
1627690337574 Marionette INFO Marionette enabled
console.info: MINT: Trying to use the system region (based on LANG).
console.info: MINT: System region is BR
console.warn: SearchSettings: "get: No settings file exists, new profile?" (new
NotFoundError("Could not open the file at /tmp/rust_mozprofileqa880X/-
search.json.mozlz4", (void 0)))
1627690342977 Marionette INFO Listening on port 41727
1627690343561 Marionette WARN TLS certificate errors will be ignored for this
session|
console.info: MINT: Loading search-config defaults.
JavaScript error: resource://gre/modules/TelemetryUtils.jsm, line 154: TypeError:
date is undefined
1627690370234 Marionette INFO Stopped listening on port 41727

```

Teste Sistema 04:

Título: Cadastro (cadastro.py)

Ferramenta: Selenium

Autor: Filipe de Oliveira Ataíde

Data da execução: 28/07/2021

```
(venv) filipe@filipe-Aspire-A315-53:~/Documentos/PDS Corporativo/cookit-api/cookit$
python manage.py test usuario.tests.CadastroTestCase.test_cadastro
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 13.210s

OK
Destroying test database for alias 'default'...
```

geckodriver.log x

```
1627690206896 geckodriver INFO Listening on 127.0.0.1:55519
1627690206901 mozrunner::runner INFO Running command: "/usr/bin/firefox" "--
marionette" "-foreground" "-no-remote" "-profile" "/tmp/rust_mozprofile4yvx1E"
1627690209425 Marionette INFO Marionette enabled
console.info: MINT: Trying to use the system region (based on LANG).
console.info: MINT: System region is BR
console.warn: SearchSettings: "get: No settings file exists, new profile?" (new
NotFoundError("Could not open the file at /tmp/rust_mozprofile4yvx1E/-
search.json.mozlz4", (void 0)))
1627690214770 Marionette INFO Listening on port 32871
1627690215133 Marionette WARN TLS certificate errors will be ignored for this
session
console.info: MINT: Loading search-config defaults.
JavaScript error: resource://gre/modules/TelemetryUtils.jsm, line 154: TypeError:
date is undefined
1627690242204 Marionette INFO Stopped listening on port 32871
```

Teste Sistema 05:

Título: Cadastro de receita (test.py)

Ferramenta: Selenium

Autor: Victor Augusto Fernandes Pereira

Link: [receita/tests.py · usuario+api · tads.cnat / pdscorporativo / 2021.1 / Cookit / cookit-api · GitLab \(ifrn.edu.br\)](https://receita.tests.py.usuario+api.tads.cnat/pdscorporativo/2021.1/Cookit/cookit-api-GitLab.ifrn.edu.br)

Data da execução: 28/07/2021

```
(project-name) λ python manage.py test receita
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
```

DevTools listening on ws://127.0.0.1:61665/devtools/browser/a28ae59a-9512-4509-92d0-0736cd8cc7c3

```
[14440:14484:0730/215915.719:ERROR:device_event_log_impl.cc(214)] [21:59:15.719] USB: usb_device_handle_win.cc:1048 Failed to read descriptor from node connection: Um d
ispositivo conectado ao sistema não está funcionando. (0x1F)
[14440:14484:0730/215915.722:ERROR:device_event_log_impl.cc(214)] [21:59:15.722] USB: usb_device_handle_win.cc:1048 Failed to read descriptor from node connection: Um d
ispositivo conectado ao sistema não está funcionando. (0x1F)
[16896:6248:0730/220013.575:ERROR:gpu_init.cc(441)] Passthrough is not supported, GL is disabled
.
-----
Ran 1 test in 236.801s

OK
Destroying test database for alias 'default'...
```

Teste Sistema 06:

Título: Teste Unitário Busca Avançada

Ferramenta: Unittest/Django Test

Autor: Alanis Isabelle de Oliveira Silva

Data da execução: 28/07/2021

```
C:\Users\alani\Documents\Estudos paralelos\Não bobos\Cookit Desmembrado\cookit-api\CookIt (usuario+api -> origin)
(venv) λ python manage.py test
Creating test database for alias 'default'...
Got an error creating the test database: database "test_cookit" already exists

Type 'yes' if you would like to try deleting the test database 'test_cookit', or 'no' to cancel: yes
Destroying old test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 13 tests in 0.342s

OK
Destroying test database for alias 'default'...
```

```

cookitapi > receita > test.py
251 # Feitos por Alanis Isabelle de Oliveira Silva
252
253 # Create your tests here. WIP
254
255 class BuscaTestCase(TestCase):
256     def setUp(self):
257         Cookinho = User.objects.create(username="CookIt", first_name="Cook", last_name="It", password="cookit")
258
259         X = Ingrediente.objects.create(nome_ingrediente="X", unidade_medida_ingrediente="U", quantidade_ingrediente=1)
260         Y = Ingrediente.objects.create(nome_ingrediente="Y", unidade_medida_ingrediente="U", quantidade_ingrediente=1)
261
262         C1 = [
263             X
264         ]
265         C2 = [
266             Y
267         ]
268
269         RA = Receita.objects.create(
270             dono_receita=Cookinho,
271             nome_receita="Biscoito",
272             modo_preparo="?",
273             porcoes=1,
274             sabor_receita="D",
275             tempo_preparo=1,
276             tempo_unidade_medida="M",
277             categoria="A",
278             dificuldade="F",
279             observacoes_adicionais="?",
280         )
281         RA.ingredientes.set(C1)
282
283         RB = Receita.objects.create(
284             dono_receita=Cookinho,
285             nome_receita="Bolo",
286             modo_preparo="?",
287             porcoes=1,
288             sabor_receita="D",
289             tempo_preparo=1,
290             tempo_unidade_medida="M",
291             categoria="B",
292             dificuldade="F",
293             observacoes_adicionais="?",
294         )
295         RB.ingredientes.set(C2)
296
297     def test_ingrediente_X(self):
298         factory = APIRequestFactory()
299         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=&categoria__in=&ingredientes__nome_ingr')
300         return request
301
302     def test_ingrediente_Y(self):
303         factory = APIRequestFactory()
304         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=&categoria__in=&ingredientes__nome_ingr')
305         return request
306
307     def test_ingrediente_nao_ter_Y(self):
308         factory = APIRequestFactory()
309         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=&categoria__in=&ingredientes__nome_ingr')
310         return request
311
312     #
313     def test_ingrediente_X_e_Y(self):
314         factory = APIRequestFactory()
315         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=&categoria__in=&ingredientes__nome_ingr')
316         return request
317
318     def test_ingrediente_Y_e_nao_ter_Y(self):
319         factory = APIRequestFactory()
320         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=&categoria__in=&ingredientes__nome_ingr')
321         return request
322
323     def test_ingrediente_X_e_nao_ter_Y(self):
324         factory = APIRequestFactory()
325         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=&categoria__in=&ingredientes__nome_ingr')
326         return request
327
328     # -----
329     def test_dificuldade_F(self):
330         factory = APIRequestFactory()
331         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=F&categoria__in=&ingredientes__nome_ingr')
332         return request
333
334     def test_dificuldade_F_nao_ter_Y(self):
335         factory = APIRequestFactory()
336         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=F&categoria__in=&ingredientes__nome_ingr')
337         return request
338
339     def test_dificuldade_F_ingrediente_X_nao_ter_Y(self):
340         factory = APIRequestFactory()
341         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=F&categoria__in=&ingredientes__nome_ingr')
342         return request
343
344     # -----
345     def test_tipo_doce(self):
346         factory = APIRequestFactory()
347         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=doce&dificuldade=F&categoria__in=&ingredientes__nome_ingr')
348         return request
349
350     def test_tipo_salgado(self):
351         factory = APIRequestFactory()
352         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=salgado&dificuldade=F&categoria__in=&ingredientes__nome_ingr')
353         return request
354
355     # -----
356     def test_categoria_B(self):
357         factory = APIRequestFactory()
358         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=&categoria__in=B&ingredientes__nome_ingr')
359         return request
360
361     def test_categoria_A_B(self):
362         factory = APIRequestFactory()
363         request = factory.get('/api/receita/?nome_receita__icontains=&sabor_receita__in=&dificuldade=&categoria__in=A%2C+B&ingredientes__nome_ingr')
364         return request
365
366

```

Teste Sistema 07:

Título: Busca Simples

Ferramenta: Selenium

Autor: Alanis Isabelle de Oliveira Silva

Data da execução: 28/07/2021

```
363
364 # -----
365 # Busca simples
366
367 class BuscaSelTestCase(TestCase):
368     def test_busca_simples_nome_X(self):
369         self.driver=Firefox()
370         self.driver.get('http://localhost:8080/')
371         self.driver.find_element_by_name('search-input').send_keys('X')
372         self.driver.find_element_by_name('search-button').click()
373
374 # -----
375
```

```
C:\Users\alani\Documents\Estudos paralelos\Não bobos\Cookit Desmembrado\cookit-api\CookIt (usuario+api -> origin)
(venv) λ python manage.py test
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 12.567s

OK
Destroying test database for alias 'default'...
```


Teste Sistema 08:

Título: Busca Avançada

Ferramenta: Selenium

Autor: Alanis Isabelle de Oliveira Silva

Data da execução: 28/07/2021

```
C:\Users\alani\Documents\Estudos paralelos\Não bobos\Cookit Desmembrado\cookit-api\CookIt (usuario+api -> origin)
(venv) λ python manage.py test
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..EEE.....
=====
ERROR: test_BuscaAvancada_dificuldade_F (receita.tests.BuscaAvancadaTestCase)
-----
Traceback (most recent call last):
  File "C:\Users\alani\Documents\Estudos paralelos\Não bobos\Cookit Desmembrado\cookit-api\CookIt\receita\tests.py", line 189, in test_BuscaAvancada_dificuldade_F
    self.driver.find_element_by_id('difficult').input(F)
  File "C:\Users\alani\AppData\Local\Programs\Python\Python39\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 360, in find_element_by_id
    return self.find_element(by=By.ID, value=id_)
  File "C:\Users\alani\AppData\Local\Programs\Python\Python39\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 976, in find_element
    return self.execute(Command.FIND_ELEMENT, {
  File "C:\Users\alani\AppData\Local\Programs\Python\Python39\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 321, in execute
    self.error_handler.check_response(response)
  File "C:\Users\alani\AppData\Local\Programs\Python\Python39\lib\site-packages\selenium\webdriver\remote\errorhandler.py", line 242, in check_response
    raise exception_class(message, screen, stacktrace)
selenium.common.exceptions.NoSuchElementException: Message: Unable to locate element: [id="difficult"]

=====
ERROR: test_BuscaAvancada_dificuldade_F_nao_ter_Y (receita.tests.BuscaAvancadaTestCase)
-----
Traceback (most recent call last):
  File "C:\Users\alani\Documents\Estudos paralelos\Não bobos\Cookit Desmembrado\cookit-api\CookIt\receita\tests.py", line 195, in test_BuscaAvancada_dificuldade_F_nao_ter_Y
    self.driver.find_element_by_id('difficult').input(F)
  File "C:\Users\alani\AppData\Local\Programs\Python\Python39\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 360, in find_element_by_id
    return self.find_element(by=By.ID, value=id_)
  File "C:\Users\alani\AppData\Local\Programs\Python\Python39\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 976, in find_element
    return self.execute(Command.FIND_ELEMENT, {
  File "C:\Users\alani\AppData\Local\Programs\Python\Python39\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 321, in execute
    self.error_handler.check_response(response)
  File "C:\Users\alani\AppData\Local\Programs\Python\Python39\lib\site-packages\selenium\webdriver\remote\errorhandler.py", line 242, in check_response
    raise exception_class(message, screen, stacktrace)
selenium.common.exceptions.NoSuchElementException: Message: Unable to locate element: [id="difficult"]

-----
Ran 13 tests in 207.442s

FAILED (errors=3)
Destroying test database for alias 'default'...
```

```

375 # Busca Avançada
376
377 class BuscaAvancadaTestCase(TestCase):
378
379     def test_BuscaAvancada_ingrediente_X(self):
380         self.driver=Firefox()
381         self.driver.get('http://localhost:8080/')
382         self.driver.find_element_by_name('querotenha').send_keys('X')
383         self.driver.find_element_by_name('bts1').click()
384
385     def test_BuscaAvancada_ingrediente_X(self):
386         self.driver=Firefox()
387         self.driver.get('http://localhost:8080/')
388         self.driver.find_element_by_name('querotenha').send_keys('X')
389         self.driver.find_element_by_name('bts1').click()
390
391     def test_BuscaAvancada_ingrediente_Y(self):
392         self.driver=Firefox()
393         self.driver.get('http://localhost:8080/')
394         self.driver.find_element_by_name('querotenha').send_keys('Y')
395         self.driver.find_element_by_name('bts1').click()
396
397     def test_BuscaAvancada_ingrediente_nao_ter_Y(self):
398         self.driver=Firefox()
399         self.driver.get('http://localhost:8080/')
400         self.driver.find_element_by_name('queronaotenha').send_keys('Y')
401         self.driver.find_element_by_name('bts2').click()
402
403     def test_BuscaAvancada_ingrediente_X_Y(self):
404         self.driver=Firefox()
405         self.driver.get('http://localhost:8080/')
406         self.driver.find_element_by_name('querotenha').send_keys('X')
407         self.driver.find_element_by_name('bts1').click()
408         self.driver.find_element_by_name('querotenha').send_keys('Y')
409         self.driver.find_element_by_name('bts1').click()
410
411     def test_BuscaAvancada_ingrediente_Y_e_nao_ter_Y(self):
412         self.driver=Firefox()
413         self.driver.get('http://localhost:8080/')
414         self.driver.find_element_by_name('querotenha').send_keys('Y')
415         self.driver.find_element_by_name('bts1').click()
416         self.driver.find_element_by_name('queronaotenha').send_keys('Y')
417         self.driver.find_element_by_name('bts2').click()
418
419     def test_ingrediente_X_e_nao_ter_Y(self):
420         self.driver=Firefox()
421         self.driver.get('http://localhost:8080/')
422         self.driver.find_element_by_name('querotenha').send_keys('X')
423         self.driver.find_element_by_name('bts1').click()
424         self.driver.find_element_by_name('queronaotenha').send_keys('Y')
425         self.driver.find_element_by_name('bts2').click()
426
427     ###
428     ####IP####
429     ...
430
431     def test_BuscaAvancada_dificuldade_F(self):
432         self.driver=Firefox()
433         self.driver.get('http://localhost:8080/')
434         self.driver.find_element_by_id('input-23').click()
435         #self.driver.execute_script("arguments[Fácil].choice;", select)
436
437     def test_BuscaAvancada_dificuldade_F_nao_ter_Y(self):
438         self.driver=Firefox()
439         self.driver.get('http://localhost:8080/')
440         self.driver.find_element_by_name('difficult').input(F)
441         self.driver.find_element_by_name('queronaotenha').send_keys('Y')
442         self.driver.find_element_by_name('bts2').click()
443
444     def test_BuscaAvancada_dificuldade_F_ingrediente_X_nao_ter_Y(self):
445         self.driver=Firefox()
446         self.driver.get('http://localhost:8080/')
447         self.driver.find_element_by_name('difficult').input(F)
448         self.driver.find_element_by_name('querotenha').send_keys('X')
449         self.driver.find_element_by_name('bts1').click()
450         self.driver.find_element_by_name('queronaotenha').send_keys('Y')
451         self.driver.find_element_by_name('bts2').click()
452
453     # -----
454     #python manage.py test receita.tests.BuscaAvancadaTestCase.test_BuscaAvancada_tipo_doce
455
456     def test_BuscaAvancada_tipo_doce(self):
457         self.driver=Firefox()
458         self.driver.maximize_window()
459         self.driver.get('http://localhost:8080/')
460         element = self.driver.find_element_by_xpath("//input[@value='D']")
461         self.driver.execute_script("arguments[0].click();", element)
462
463     def test_BuscaAvancada_tipo_salgado(self):
464         self.driver=Firefox()
465         self.driver.get('http://localhost:8080/')
466         element = self.driver.find_element_by_xpath("//input[@value='S']")
467         self.driver.execute_script("arguments[0].click();", element)
468
469     # -----
470
471     def test_BuscaAvancada_categoria_B(self):
472         self.driver=Firefox()
473         self.driver.get('http://localhost:8080/')
474         element = self.driver.find_element_by_xpath("//input[@value='B']")
475         self.driver.execute_script("arguments[0].click();", element)
476
477     def test_BuscaAvancada_categoria_A_B(self):
478         self.driver=Firefox()
479         self.driver.get('http://localhost:8080/')
480         element1 = self.driver.find_element_by_xpath("//input[@value='A']")
481         self.driver.execute_script("arguments[0].click();", element1)
482         element2 = self.driver.find_element_by_xpath("//input[@value='B']")
483         self.driver.execute_script("arguments[0].click();", element2)
484

```

Testes de Integração

Título: Login e Cadastro

Ferramenta: Selenium

Autores: Alanis, Filipe, Thales, Victor

Data da execução: 28/09/2021

```
class IntegracaoTestCase(TestCase):
    def test_cadastro(self):
        self.driver = Firefox()
        self.driver.get("http://localhost:8080/Cadastro")

        self.driver.find_element_by_name("first_name").send_keys("Teste")
        self.driver.find_element_by_name("last_name").send_keys("CookIt")
        self.driver.find_element_by_name("username").send_keys("testecookit")
        self.driver.find_element_by_name("email").send_keys("teste@cookit.com")
        self.driver.find_element_by_name("password").send_keys("re12345678")
        self.driver.find_element_by_name("re_password").send_keys("re12345678")
        self.driver.find_element_by_name("submit_form").click()

        self.driver.get("http://localhost:8080/Login")

        self.driver.find_element_by_name("email").send_keys("teste@cookit.com")
        self.driver.find_element_by_name("password").send_keys("re12345678")
        self.driver.find_element_by_name("submit_form").click()
```