		Comentário
1. Por que as práticas de design de teste não podem ser aplicadas imediatamente após o recebimento dos requisitos?		
As práticas de design de teste não podem começar logo de cara depois de receber os requisitos porque a gente precisa de um tempo para entender tudo direitinho e planejar os testes. Primeiro, é necessário ler os requisitos com calma, ver se eles estão confusos, ambíguos ou incompletos, e, claro, entender bem como o sistema vai funcionar e como os usuários vão usá-lo. Também é importante decompor esses requisitos em partes menores e mais fáceis de gerenciar, para garantir que cada detalhe seja testado da forma certa. Se aparecerem áreas cinzentas ou vagas durante esse processo, é essencial conversar com a equipe para esclarecer e evitar mal-entendidos. Além disso, a gente precisa definir uma estratégia de teste e garantir que temos os recursos necessários. Preparar o ambiente de teste e configurar as ferramentas também é fundamental. Esse tempo de preparação é super importante para que os testes sejam eficazes e o software saia sem falhas. Se a gente ignorar essa análise e já começar o design do teste logo de cara, pode acabar deixando passar falhas importantes, o que prejudicaria a qualidade final do produto.		
2. Em que situações as classes de equivalência e os valores-limite podem existir separadamente? Explique sua resposta e dê exemplos.		
Apesar de geralmente nós utilizarmos as classes de equivalência e valores-limite juntos para fazer testes mais completos, às vezes podemos usá-los separadamente, dependendo do que estamos tentando testar e do que precisamos verificar. Abaixo vou explicar o que seria cada um e, depois, exemplicar como podemos utilizar de forma separada. **O que são Classes de Equivalência?** Classes de equivalência são valores processados da mesma forma por um aplicativo ou sistema. *Exemplo:* Se um campo de idade aceita valores de 1 a 100, podemos ter essas classes de equivalência: Valores válidos: 1 a 100 Valores inválidos (negativos): qualquer número menor que 1 Valores inválidos (acima do limite): qualquer número maior que 100 Caracteres especiais: @/%^! Uso de espaço entre os número: 1 9 Dentre outros.... **O que são Valores-Limite?** Os valores-limite testam os limites das classes de equivalência para garantir que o sistema lide corretamente com as transições entre as classes. *Exemplo:* Vou utilizar o mesmo exemplo que usei em Classe de Equivalência (idade). Limite inferior: 1 (válido) e 0 (inválido) Limite superior: 100 (válido) e 101 (inválido) **Situações onde podem existir separadamente?** ***Classes de Equivalência Sem Valores-Limite:*** Quando testamos funcionalidades onde os limites não são tão críticos, mas queremos garantir que diferentes tipos de entradas são aceitas. *Exemplo:* Um campo de seleção de gênero com as opções "Masculino", "Feminino" e "Outro". Aqui, as classes de equivalência seriam os três valores possíveis. Não há valores-limite a serem considerados. ***Valores-Limite Sem Classes de Equivalência:*** Quando estamos mais interessados em verificar como o sistema se comporta nos extremos dos valores permitidos, independentemente das classes de equivalência. *Exemplo:* Testar um controle de volume de um dispositivo que vai de 0 a 10. Podemos nos concentrar apenas em verificar se o controle lida corretamente com os valores 0 e 10, sem preocupar-nos com todas as classes de equivalência intermediárias.		
3. O que é equivalência? O que é uma classe de equivalência?		
Equivalência significa que diferentes entradas ou condições são tratadas da mesma forma pelo sistema. Ou seja, se duas entradas são consideradas iguais para o sistema, elas devem ter o mesmo resultado. **Classe de equivalência** é um grupo de entradas que o sistema trata da mesma maneira. Em vez de testar todas as possíveis entradas, você testa uma entrada de cada grupo para garantir que o sistema funciona bem para todas as entradas desse grupo. *Exemplo:* **Equivalência:** Se um campo aceita números de 1 a 10, o número 6 e o número 8 são tratados igual porque ambos estão dentro do intervalo permitido. **Classe de equivalência:** Para o campo de números, as classes seriam: Valores válidos: qualquer número de 1 a 10 (como 6 e 8) Valores inválidos: números fora desse intervalo (como -1 e 11) Essas classes ajudam a testar o sistema de forma mais eficiente, cobrindo todos os tipos de entradas sem precisar testar cada valor individualmente.		
4. É possível excluir verificações no meio do intervalo em favor de verificações nos limites dentro do intervalo? Explique sua resposta.		
Sim, é possível pular os testes no meio do intervalo e focar apenas nos limites, porque muitos problemas aparecem nas bordas. Testar os limites e alguns valores ao redor geralmente é suficiente e reduz o número de testes necessários. Por exemplo: Para números de 1 a 10, você pode testar os limites 1 e 10, e valores próximos como 0 e 11. Para idades de 18 a 65, teste 18, 65, e valores ao redor como 17 e 66. Porém, não é recomendado ignorar completamente as verificações no meio do intervalo. Apesar de os problemas aparecerem com mais frequência nos limites, testar alguns valores intermediários ajuda a garantir que o sistema funcione bem em todas as situações, evitando falhas inesperadas. Focar nos limites é uma maneira prática de testar, mas é importante encontrar um equilíbrio e não deixar de lado outros pontos importantes.		
5. Imagine que você precise testar um formulário onde cada campo possui um validador. O resultado do trabalho do formulário depende da combinação de dados nos campos. Quais práticas de design de teste devem ser usadas e por quê? Explique sua resposta.		

Para testar um formulário com vários campos e validadores, as seguintes práticas de design de teste são essenciais:

Classes de Equivalência: Essa técnica ajuda a identificar diferentes grupos de entradas que o sistema deve tratar da mesma forma. Por exemplo, se um campo aceita idades entre 18 e 65 anos, você pode criar classes de equivalência para idades válidas (18 a 65), idades inválidas abaixo de 18, e idades inválidas acima de 65. Isso reduz o número de testes, focando apenas nos grupos relevantes de entradas.

Valores-Limite: Erros geralmente ocorrem nos limites das entradas válidas. Por exemplo, se um campo aceita números de 1 a 100, você deve testar os limites 1 e 100, além de valores próximos, como 0 e 101. Isso garante que o formulário está lidando corretamente com entradas no extremo do intervalo permitido.

Tabelas de Decisão: Usar tabelas de decisão é fundamental quando o resultado do formulário depende de combinações de dados nos campos. Essas tabelas permitem mapear as diferentes combinações de entradas e os resultados esperados. Por exemplo, em um formulário de cadastro que exige um e-mail e uma senha, uma tabela de decisão pode ajudar a testar cenários onde o e-mail está correto, mas a senha é inválida, e vice-versa.

Testar o Formulário Todo: Preencher todos os campos e enviar o formulário verifica se todas as partes funcionam juntas corretamente. Isso ajuda a identificar problemas que podem não ser visíveis ao testar apenas partes individuais.

Facilidade de Uso: Além dos testes funcionais, é importante garantir que o formulário seja fácil de usar. Isso inclui testar a clareza das mensagens de erro e a navegabilidade do formulário.

Essas práticas ajudam a garantir que o formulário seja testado de maneira eficiente, cobrindo diferentes cenários e combinações, além de oferecer uma boa experiência ao usuário.

6. Explique como você pode otimizar o número de testes ao trabalhar com tabelas de decisão. Explique sua resposta.

Para otimizar os testes usando tabelas de decisão, começamos listando as condições (como idade, renda, etc.) e as ações que o sistema deve realizar (como aprovar ou rejeitar). Cada condição pode ter diferentes estados, e cada combinação dessas condições pode levar a uma ação específica. Depois, montamos uma tabela onde cada linha mostra uma combinação de condições e o resultado correspondente. Isso ajuda a visualizar todas as possibilidades e a criar casos de teste para cada combinação.

Para otimizar o número de testes, podemos fazer o seguinte:

Agrupar Combinações Semelhantes: Analisamos a tabela para identificar combinações que levam ao mesmo resultado. Se várias combinações produzem a mesma ação, agrupamos essas combinações e testamos apenas uma que represente o grupo. Isso reduz a quantidade de testes repetitivos.

Simplificar as Condições: Se uma condição não afeta o resultado em alguns casos, podemos simplificá-la ou ignorá-la nesses cenários. Por exemplo, se a "renda" não importa para alguém maior de 65 anos, não precisamos testar a renda nesses casos.

Criar Casos de Teste Representativos: Após simplificar a tabela, escolhemos casos de teste que cubram todas as ações possíveis, mas com o menor número de combinações. Cada caso de teste é selecionado para garantir que todas as ações importantes do sistema sejam verificadas.

Eliminar Casos Redundantes: No final, revisamos a tabela para remover casos que não trazem novos insights e só repetem o que já foi testado.

Por exemplo, se um formulário aprova uma inscrição para idades entre 18 e 65 anos, não é necessário testar cada idade individualmente. É suficiente testar os limites (18 e 65) e alguns valores intermediários, como 30 ou 50, para garantir que o sistema funcione bem para toda a faixa.

Seguindo esses passos, conseguimos cobrir todas as ações importantes do sistema sem precisar fazer testes demais, economizando tempo e esforço.

7. Descreva como um checklist difere dos casos de teste. Dê exemplos de onde ambos são usados.

Um checklist e casos de teste são usados para garantir que o software funciona bem, mas de maneiras diferentes.

Um checklist é uma lista de coisas para verificar, de forma geral. Por exemplo, checar se todos os botões funcionam, se as imagens carregam e se o texto está correto. É usado para garantir que todas as áreas principais sejam verificadas.

Casos de teste são descrições detalhadas de como testar funcionalidades específicas. Eles incluem entradas, ações e resultados esperados. Por exemplo, testar se o login funciona com credenciais válidas ou inválidas, e se a recuperação de senha está correta.

A diferença principal é que checklists são mais gerais e rápidos, enquanto casos de teste são detalhados e específicos. Checklists garantem que tudo foi checado, enquanto casos de teste verificam se cada função específica está funcionando.

8. Qual é a maneira correta de fazer um relatório de bug? Quais elementos de um relatório de bug são necessários? Por quê?

Para fazer um relatório de bug corretamente, é importante incluir os seguintes elementos:

ID do Bug: Um código único para identificar o bug.
Título: Um resumo curto e claro do problema.
Prioridade e Severidade: Indicam a urgência e o impacto do bug.
Ambiente do Sistema: Informações sobre o sistema operacional, navegador ou versão do aplicativo onde o bug foi encontrado.
Passos para Reproduzir: Instruções detalhadas para recriar o bug.
Resultados Reais e Esperados: Descrição do que aconteceu versus o que deveria ter acontecido.
Evidências: Capturas de tela, vídeos ou logs que ajudem a ilustrar o problema.

Esses elementos são necessários para que a equipe de desenvolvimento possa entender o bug, reproduzi-lo e corrigi-lo de maneira eficiente.

9. Quais são as regras para escrever um título de relatório de bug? O que acontece se você criar um título ruim?

Para escrever um bom título de relatório de bug, precisamos ser claros e diretos, usando palavras-chave importantes. Se o título for ruim, fica difícil achar o bug depois, entender o problema rapidamente ou saber o quão urgente ele é. Isso pode atrasar a correção e causar confusão na equipe. Para escrever um título, é preciso fazer três perguntas: "O quê? Onde? Quando?" Ao mesmo tempo, também tentamos descrever o resultado real (em vez do esperado). Um bom título ajuda todo mundo a entender, encontrar e corrigir o problema mais rápido.

10. Em que consiste a arquitetura de um aplicativo cliente-servidor? Descreva resumidamente a funcionalidade de cada elemento.

Na arquitetura de um aplicativo cliente-servidor, temos os seguintes componentes principais:

Cliente: É a parte com a qual o usuário interage. O cliente é onde você insere dados e faz solicitações. Ele envia essas solicitações para o servidor e mostra os resultados de volta para você.

Servidor: Recebe as solicitações do cliente, processa as informações e acessa os dados necessários. Depois, o servidor manda a resposta de volta para o cliente. O servidor cuida do processamento e do gerenciamento de dados.

Protocolo de Comunicação: Define como o cliente e o servidor trocam informações. Protocolos como HTTP (para a web) e TCP/IP (para redes) garantem que as mensagens sejam enviadas e recebidas corretamente.

Banco de Dados: Armazena as informações que o servidor precisa para responder às solicitações. O servidor acessa e atualiza o banco de dados quando necessário e depois envia as informações processadas de volta ao cliente.

Resumindo, o cliente cuida da interação com o usuário, o servidor faz o processamento e gerencia os dados, o protocolo de comunicação garante que as informações sejam trocadas de forma correta, e o banco de dados armazena as informações necessárias. A comunicação entre esses componentes é essencial para que o aplicativo funcione bem.

11. Descreva as etapas de processamento da solicitação após esta URL ser inserida na barra de endereços do navegador: https://google.com.

Quando digitamos uma URL, como https://google.com, na barra de endereços, o navegador começa enviando uma solicitação para um servidor DNS. O servidor DNS traduz o nome do domínio em um endereço IP, que é o identificador exclusivo do servidor que tem as informações que queremos. Com o endereço IP em mãos, o navegador se conecta ao servidor usando esse IP.

Depois, o navegador estabelece uma conexão segura (já que estamos usando https), e então envia uma solicitação para o servidor pedindo a página inicial do Google. O servidor processa a solicitação e retorna a página web para o navegador. Finalmente, o navegador exibe a página principal do Google na nossa tela, pronta para ser usada.

12. O que é um cache? Por que é necessário? Qual regra deve ser seguida ao trabalhar com um cache no teste?

Um cache é basicamente um espaço onde guardamos dados temporariamente para acessá-los mais rápido depois. Quando pedimos algo, o sistema confere primeiro no cache antes de buscar na fonte original, que pode ser mais lenta. Usamos cache para deixar tudo mais rápido e eficiente, ajudando a acelerar o acesso aos dados, reduzir a carga no servidor e permitir que o sistema suporte mais usuários sem travar.

Quando estamos testando sistemas que usam cache, precisamos ter certeza de que o cache não vai atrapalhar nossos testes. Para isso, devemos limpar o cache antes de começar os testes para garantir que estamos usando dados atualizados. Também é importante verificar se o cache está funcionando corretamente e se atualiza os dados como deveria. Além disso, precisamos testar com dados variados para garantir que o cache está sendo atualizado corretamente e não está causando problemas.

13. Responda às perguntas sobre os protocolos HTTP e HTTPS:
- Qual é a diferença entre HTTP e HTTPS? Quando você não deve usar HTTP?
- Quais componentes compõem uma solicitação HTTP: para que cada um serve?
- Quais métodos HTTP você conhece? Para que eles servem? Dê exemplos de aplicação de diferentes métodos.
- O que é um código de resposta HTTP? Quais códigos existem?

A principal diferença entre HTTP e HTTPS é que o HTTPS é a versão segura do HTTP. O HTTPS usa criptografia para proteger os dados que trocamos entre o navegador e o servidor, o que ajuda a evitar que informações sejam interceptadas ou manipuladas por terceiros. Usamos HTTPS sempre que lidamos com informações sensíveis, como dados de login ou dados financeiros. Não devemos usar HTTP em situações onde a segurança é importante, porque os dados são enviados em texto claro e podem ser facilmente interceptados.

Uma solicitação HTTP é composta por vários componentes:

Método: Indica a ação que queremos realizar, como obter ou enviar dados. Exemplos são GET, POST, PUT e DELETE.
URL (Uniform Resource Locator): Especifica o endereço do recurso que queremos acessar.
Cabeçalhos: Fornecem informações adicionais sobre a solicitação, como tipo de conteúdo aceito e informações de autenticação.
Corpo (Body): Contém os dados enviados ao servidor, geralmente usado em métodos como POST e PUT.

Alguns métodos HTTP comuns são:

GET: Usado para solicitar dados de um servidor. Por exemplo, quando acessamos uma página da web.
POST: Usado para enviar dados ao servidor. Por exemplo, ao preencher um formulário.
PUT: Usado para atualizar um recurso existente. Por exemplo, atualizar as informações de perfil em um site.
DELETE: Usado para excluir um recurso. Por exemplo, deletar um comentário em um blog.

Os códigos de resposta HTTP informam o status da solicitação. Alguns exemplos são:

200 OK: A solicitação foi bem-sucedida e o servidor retornou a resposta esperada.
404 Not Found: O recurso solicitado não foi encontrado no servidor.
500 Internal Server Error: Ocorreu um erro no servidor ao processar a solicitação.
301 Moved Permanently: O recurso foi movido permanentemente para um novo URL.
403 Forbidden: O servidor entendeu a solicitação, mas se recusa a autorizá-la.

14. Descreva os componentes que podem compor uma URL. Pelo que ele é responsável?

- Esquema: Protocolo usado para transmitir dados. Ex.: http:// https://

- Nome de usuário e senha: Informações sobre o usuário que solicitou os dados. Ex.: miguel:thebestqa

- Nome do host: Endereço exclusivo do aplicativo web. Ex.: google.com

- Porta: Tipo específico de conexão. Na maioria das vezes, você não precisa inseri-la: o servidor usa automaticamente a porta correta. Ex.: 443

- Caminho: Endereço de uma página específica em um aplicativo web. Ele é separado por barras /. Ex.: /search/: a página de pesquisa /profile/: o perfil do usuário

- Parâmetros da solicitação: Informações adicionais enviadas ao servidor, por exemplo, para solicitar dados específicos. Eles são listados depois do sinal de ? e separados com &. Ex.: https://www.google.com/search?q=figma: q=figma significa que você está pesquisando pela palavra figma no Google

- Âncora: Indicador adicional que permite chegar diretamente à parte necessária de uma página web, por exemplo, ao título ou parágrafo de um texto. Ela é útil em páginas longas com múltiplas seções. Ex.: https://developer.chrome.com/docs/devtools/network/#overview: a #overview âncora leva o usuário diretamente à seção ""Visão Geral"" da página. Isso é bem útil em páginas longas com múltiplas seções."

15. Verifique se há erros no JSON
Se houver, anexe a versão correta do arquivo.

Link para a ilustração:

https://practicum-content.s3.us-west-1.amazonaws.com/new-markets/qa-final-project/US/7.1.1.png

Consegui encontrar 7 erros no JSON.
Segue versão correta abaixo:

```
{
    "Menu": {
        "id": "1",
        "value": "File",
        "list": {
            "items": {
                "new_doc": {
                    "value": "New",
                    "onclick": "create_new_doc"
                },
                "open_doc": {
                    "value": "Open...",
                    "onclick": "open_doc"
                },
                "save_doc": {
                    "value": "Save",
                    "onclick": "save_doc"
                },
                "save_as_doc": {
                    "value": "Save as...",
                    "onclick": "save_as_doc"
                },
                "print_option": {
                    "value": "Print options",
                    "onclick": "21",
                    "show_print_option": {
                        "Color": "Saturated",
                        "Black and white printing?": "",
                        "Print Size": "A4"
                    }
                }
            }
        }
    }
}
```

16. O que é um banco de dados relacional? Como ele difere de um não relacional?

Um banco de dados relacional organiza dados em tabelas com linhas e colunas, como uma planilha do Excel. A gente usa SQL (Structured Query Language) para consultar e manipular esses dados. As tabelas podem se conectar umas com as outras usando chaves, o que ajuda a manter os dados organizados e relacionados.

Exemplos de bancos de dados relacionais:

MySQL: Um sistema popular que usa SQL para gerenciar dados estruturados.

PostgreSQL: Outro exemplo de banco de dados relacional conhecido por sua robustez e conformidade com o SQL.

Exemplos de bancos de dados não relacionais:

MongoDB: Armazena dados em documentos no formato JSON, ideal para dados que não têm um esquema fixo e podem variar.

Redis: Um banco de dados em memória que usa pares chave-valor, ótimo para caching e armazenamento temporário.

A principal diferença entre bancos de dados relacionais e não relacionais está na forma como os dados são organizados e manipulados:

Relacional: Tem uma estrutura mais rígida e organizada em tabelas. É ótimo para dados bem definidos e organizados, como em sistemas financeiros ou lojas online. Por exemplo, você pode ter uma tabela para "Clientes" e outra para "Pedidos", e as tabelas se conectam através de chaves, como um ID de cliente que vincula os pedidos a um cliente específico.

Não Relacional: É mais flexível e pode armazenar dados de várias maneiras, como documentos (JSON, XML), pares chave-valor, ou grafos. Isso é útil para dados que mudam com frequência ou que não têm uma estrutura fixa. Por exemplo, bancos de dados não relacionais como MongoDB (documento), Redis (chave-valor) e Neo4j (grafo) são usados para redes sociais, análise de grandes volumes de dados e sistemas de recomendação.

Em resumo, usamos bancos de dados relacionais quando os dados são bem organizados e precisam de relacionamentos claros. Optamos por bancos de dados não relacionais quando precisamos de mais flexibilidade e escalabilidade para lidar com diferentes tipos de dados.

Descrição da revisão	Google Chrome 1280x720	Opera 71 1280x720	Link para o relatório de bug
Cabeçalho			
O botão "concordar" é exibido para que o usuário concorde com o uso de cookies.	APROVADO	APROVADO	
O botão "status do pedido" é exibido com fundo branco e forte preta.	APROVADO	APROVADO	
Ao clicar no botão "status do pedido" é aberto uma área "insira o número do pedido.	APROVADO	APROVADO	
O botão "Fazer pedido" é exibido no campo superior direito com fundo preto e fonte branca.	APROVADO	APROVADO	
Um desenho técnico de uma scooter é exibida ao lado direito da página.	APROVADO	APROVADO	
O título "Urban Scooter"é exibido no canto superior esquerdo.	APROVADO	APROVADO	
O subtítulo "PLATAFORMA INTERATIVA" é exibido ao lado de "Urban Scooter" no canto superior esquerdo.	APROVADO	APROVADO	
O título "Scooter para dois dias" é exibido no canto superior esquerdo.	APROVADO	APROVADO	
O subtítulo "Nós levaremos até sua porta e, quando terminar, nós buscaremos" é exibido abaixo de "Scooter para dois dias"' no canto superior esquerdo.	APROVADO	APROVADO	
Quando o usuário rola pela tela, ocorre uma animação: o desenho é substituído por uma foto de uma scooter e aparece uma tabela com a descrição da scooter.	APROVADO	APROVADO	
A tabela com a descrição da scooter é exibida conforme os requisitos.	APROVADO	APROVADO	
Quando o usuário roda pela tela, o subtítulo "Nós levaremos até sua porta e, quando terminar, nós buscaremos" é substituido por "Ela é leve e tem rodas potentes, perfeitas para andar na orla e manter suavemente os dedos no guidão.	APROVADO	APROVADO	
Se o usuário rolar a tela novamente, verá a frase "Como funciona" no canto superior esquerdo.	APROVADO	APROVADO	
Um descritivo com 4 etapas de "Como funciona" é exibido conforme os requisitos.	APROVADO	APROVADO	
O botão "Fazer pedido" é exibido no campo inferior central com fundo preto e fonte branca.	APROVADO	APROVADO	
Se o usuário rolar a tela mais uma vez, verá a frase "Perguntas frequentes" no canto superior esquerdo.	APROVADO	APROVADO	
7 perguntas de "Perguntas frequentes" é exibido conforme os requisitos.	APROVADO	APROVADO	
Ao clicar no título "Urban Scooter" o usuário retorna para a página inicial.	APROVADO	APROVADO	
Dados de pedido do usuário			
Ao clicar em "Fazer pedido" o campo de preenchimento "Para quem é a scooter" aparece conforme os requisitos.	APROVADO	APROVADO	
Os campos pedem as seguintes informações: "Nome", "Sobrenome", "Endereço: para onde levar a scooter","Estação de metrô" e "Telefone: o entregador ligará para este número".	APROVADO	APROVADO	
Todos os campos de "Para quem é a scooter" são obrigatórios. Se não forem preenchidos corretamente, o usuário não vai poder passar para a próxima página.	APROVADO	APROVADO	
Há um botão "Avançar" na parte inferior: ele leva o usuário ao formulário de "Locação".	APROVADO	APROVADO	
No formulário os campos são: "Quando entregar a scooter", "Período de locação", "Cor" e "Comentário".	APROVADO	APROVADO	
Os campos "Quando entregar a scooter" e "Período de locação" são obrigatórios. "Cor" e "Comentário" são opcionais.	APROVADO	APROVADO	
Quando o botão "Voltar" é clicado o usuário é direcionado para a página "Para quem é a scooter".	APROVADO	APROVADO	
Quando o usuário alterna entre páginas, as informações inseridas são salvas.	APROVADO	APROVADO	
O botão "Voltar" é exibido em fundo branco com fonte preta.	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-149
O botão "Fazer pedido" é exibido em fundo preto com fonte branca.	APROVADO	APROVADO	
Quando o usuário clica no botão "Fazer pedido" o pedido será feito e uma janela pop-up vai aparecer com as seguintes informações: - O pedido foi feito - "Número do pedido NNNNN" - "Anote esse número: ele vai ser útil para acompanhar o status" - Um botão "Verificar status" no lado inferior direito também é exibido.	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-150
Cadeia de status do pedido			
Se o usuário clicar em "Status do pedido" no cabeçalho da página de destino, o campo "Insira o número do pedido" será exibido.	APROVADO	APROVADO	
O usuário precisa inserir um valor no campo "Insira o número do pedido" e pressionar Enter.	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-151
O usuário precisa inserir um valor no campo "Insira o número do pedido" e clique no no botão "Ir!"	APROVADO	APROVADO	
Se o número do pedido for inserido corretamente as informações "nome, sobrenome, endereço e outros" serão exibidas em um espaçamento.	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-152
Se o número do pedido for inserido corretamente, estas informações serão exibidas: Dados do pedido do usuário: nome, sobrenome, endereço e outros.	APROVADO	APROVADO	
Existe uma regra para todos os campos: se o texto não couber em uma linha, ele será transferido para a segunda linha.	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-153
Validar campo "Quando entregar a scooter"	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-160
O status atual é destacado em preto, os outros ficam em cinza.	APROVADO	APROVADO	
Se o status for aprovado, o número dele muda para um sinal de concluído.	APROVADO	APROVADO	
Se o número do pedido for inserido incorretamente, aparece uma mensagem de erro: "Esse pedido não existe. Tem certeza de que o número está correto?".	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-154
Se o número do pedido for inserido incorretamente aparece uma imagem de uma scooter no chão conforme os requisitos.	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-155
Validar o status: "A scooter está no depósito."	APROVADO	APROVADO	
Validar o status: "O entregador está a caminho."	APROVADO	APROVADO	
Validar o status: "O entregador chegou."	BLOQUEADO	BLOQUEADO	Não há endpoint para realizar o teste
Validar o status: "Ok, vamos dar uma volta."	BLOQUEADO	BLOQUEADO	Não há endpoint para realizar o teste
O "firstName" do entregador sempre é exibido completo no design.	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-157
Validar período de locação: Um dia	BLOQUEADO	APROVADO	Não foi possível realizar esse teste pelo aplicativo web no Chrome
Validar período de locação: Dois dias	BLOQUEADO	APROVADO	Não foi possível realizar esse teste pelo aplicativo web no Chrome
Validar período de locação: Três dias	BLOQUEADO	APROVADO	Não foi possível realizar esse teste pelo aplicativo web no Chrome
Validar período de locação: Quatro dias	BLOQUEADO	APROVADO	Não foi possível realizar esse teste pelo aplicativo web no Chrome
Validar período de locação: Cinco dias	BLOQUEADO	APROVADO	Não foi possível realizar esse teste pelo aplicativo web no Chrome
Validar período de locação: Seis dias	BLOQUEADO	APROVADO	Não foi possível realizar esse teste pelo aplicativo web no Chrome
Validar período de locação: Sete dias	BLOQUEADO	APROVADO	Não foi possível realizar esse teste pelo aplicativo web no Chrome
Cancelar pedido			
Ao criar um pedido o botão "Cancelar pedido" fica disponível.	APROVADO	APROVADO	
Se o usuário clicar em "Cancelar pedido", uma janela pop-up aparece com o texto "Deseja cancelar o pedido?"	APROVADO	APROVADO	
Existem dois botões na janela pop-up: "Cancelar" e "Voltar".	APROVADO	APROVADO	
Se o usuário clicar em "Voltar", ele retorna à página de status do pedido.	APROVADO	APROVADO	
Se o usuário clicar em "Cancelar pedido", uma janela pop-up aparece com o texto "Pedido cancelado. Sinta-se à vontade para voltar quando quiser :)"	APROVADO	APROVADO	
Abaixo da frase "Pedido cancelado. Sinta-se à vontade para voltar quando quiser :)", há um botão "Ok" no canto inferior direito. O botão "Ok" leva o usuário à página de destino.	APROVADO	APROVADO	
O pedido cancelado é excluído do sistema. O usuário não vai poder mais visualizá-lo.	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-156
O usuário pode cancelar o pedido até o momento em que o entregador pega a scooter.	APROVADO	APROVADO	

Quando o pedido já estiver com o entregador, o botão "Cancelar pedido" vai deixar de ser clicável.	APROVADO	APROVADO						
Pedido em atraso								
Status 1 "A scooter está no depósito.": Se o pedido estiver atrasado, o seu status muda para "O entregador está atrasado" e o texto muda para "Não será possível entregar a scooter a tempo. Para esclarecer o status do seu pedido, ligue para o suporte: 0101."	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-158					
Status 2 "O entregador está a caminho." : Se o pedido estiver atrasado, o seu status muda para "O entregador está atrasado" e o texto muda para "Não será possível entregar a scooter a tempo. Para esclarecer o status do seu pedido, ligue para o suporte: 0101."	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-159					
Status 3 "O entregador chegou.": Se o pedido estiver atrasado, o seu status muda para "O entregador está atrasado" e o texto muda para "Não será possível entregar a scooter a tempo. Para esclarecer o status do seu pedido, ligue para o suporte: 0101."	BLOQUEADO	BLOQUEADO	Não há endpoint para realizar o teste					
Status 4 "Ok, vamos dar uma volta.": Se o pedido estiver atrasado, o seu status muda para "O entregador está atrasado" e o texto muda para "Não será possível entregar a scooter a tempo. Para esclarecer o status do seu pedido, ligue para o suporte: 0101."	BLOQUEADO	BLOQUEADO	Não há endpoint para realizar o teste					
O status e a legenda são destacados em vermelho.	APROVADO	APROVADO						
O status do pedido muda para "O entregador está atrasado" e o texto muda para "Não será possível entregar a scooter a tempo" se o pedido não for entregue até as 23h59 do dia da entrega.	APROVADO	APROVADO						

Campo	Nome da classe	Limites	Dados de teste dentro da classe (conteúdo do campo)	Dados de teste nos limites (campo de comentários)	Google Chrome 1280x720	Opera 71 1280x720	Link para o relatório de bug
Nome	O comprimento deve ser de 2 a 15 caracteres.	2,15	elaine (6)	e (1) el (2) ela (3) elaineeeeeeeee (14) elaineeeeeeeeee (15) elaineeeeeeeeeee (16)	APROVADO	APROVADO	
	Menor que 2 caracteres	1	~~e (1)~~	~~e (1)~~ ~~el (2)~~	APROVADO	APROVADO	
	Maior que 15 caracteres	16	elaineeeeeeeeeeeee (18)	~~elaineeeeeeeeee (15)~~ ~~elaineeeeeeeeeee (16)~~ elaineeeeeeeeeee (17)	APROVADO	APROVADO	
	Letras latinas		elaine		APROVADO	APROVADO	
	Letras não latinas		정국		APROVADO	APROVADO	
	Nome com hífen		elaine-		APROVADO	APROVADO	
	Nome com espaço		elaine mateus		APROVADO	APROVADO	
	Nome com números		elaine123		APROVADO	APROVADO	
	Nome com caracteres especiais		elaine@		APROVADO	APROVADO	
	Campo vazio		[]		APROVADO	APROVADO	
Sobrenome	O comprimento deve ser de 2 a 15 caracteres.	2,15	mateus (6)	m (1) ma (2) mat (3) mateusssssssss (14) mateusssssssss (15) mateusssssssssss (16)	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-108
	Menor que 2 caracteres	1	~~m (1)~~	~~m (1)~~ ~~ma (2)~~	APROVADO	APROVADO	
	Maior que 15 caracteres	16	mateusssssssssssss (18)	~~mateusssssssssse (15)~~ ~~mateusssssssssse (16)~~ mateusssssssssss (17)	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-109 https://elainemtaraujo.atlassian.net/browse/BR-
	Letras latinas		mateus		APROVADO	APROVADO	
	Letras não latinas		정국		APROVADO	APROVADO	
	Nome com hífen		mateus-		APROVADO	APROVADO	
	Nome com espaço		mateus araujo		APROVADO	APROVADO	
	Nome com números		mateus123		APROVADO	APROVADO	
	Nome com caracteres especiais		mateus@		APROVADO	APROVADO	
	Campo vazio		[]		APROVADO	APROVADO	
Endereço	O comprimento deve ser de 5 a 50 caracteres.	5,50	1528 Broadway, Times Square, New York, NY 10036 (47)	1528 (4) 1528B (5) 1528 B (6) 1528 Broadway, Times Square, New York, NY 1003666 (49) 1528 Broadway, Times Square, New York, NY 10036666 (50) 1528 Broadway, Times Square, New York, NY 100366666 (51)	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-111
	Menor que 5 caracteres	4	15 (2)	152 (3) ~~1528 (4)~~ ~~1528B (5)~~	APROVADO	APROVADO	
	Maior que 50 caracteres	49	~~1528 Broadway, Times Square, New York, NY 10036 (47)~~	1528 Broadway, Times Square, New York, NY 100366 (48) ~~1528 Broadway, Times Square, New York, NY 1003666 (49)~~ ~~1528 Broadway, Times Square, New York, NY 10036666 (50)~~	APROVADO	APROVADO	
	Letras latinas		Broadway		APROVADO	APROVADO	
	Letras não latinas		정국		APROVADO	APROVADO	
	Endereço com barra		Broadway/		REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-112
	Endereço com vírgula		Broadway,		APROVADO	APROVADO	
	Endereço com ponto		Broadway.		APROVADO	APROVADO	
	Endereço com hífen		Broadway-		APROVADO	APROVADO	
	Endereço com espaço		1528 Broadway		APROVADO	APROVADO	
	Endereço com números		1528Broadway		APROVADO	APROVADO	
	Endereço com caracteres especiais		Broadway@		APROVADO	APROVADO	
	Campo vazio		[]		REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-113
Estação do metro	Estação existente		1st Street		APROVADO	APROVADO	
	Estação não existente		Morango		APROVADO	APROVADO	
	Campo vazio				APROVADO	APROVADO	
	Campo obrigatório		~~1st Street~~		APROVADO	APROVADO	
Telefone	O comprimento deve ser de 10 a 12 caracteres.	10,12	85999912047 (11)	859999120 (9) 8599991204 (10) ~~8599912047 (11)~~ 85999120477 (12) 8599991204777 (13)	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-114 https://elainemtaraujo.atlassian.net/browse/BR-115
	Menor que 10 caracteres	9	859999 (6)	85999912 (8) ~~859999120 (9)~~ ~~8599991204 (10)~~	APROVADO	APROVADO	
	Maior que 12 caracteres	13	8599991204777777 (16)	~~8599991204777 (12)~~ ~~8599991204777 (13)~~ 8599991204777 (14)	APROVADO	APROVADO	
	Telefone com letras latinas		85999912047e		APROVADO	APROVADO	
	Telefone com letras não latinas		85999912047정		APROVADO	APROVADO	
	Telefone com símbolo "+"		+85999912047		APROVADO	APROVADO	
	Telefone com espaço		8599991 2047		APROVADO	APROVADO	
	Telefone com caracteres especiais		85999912047@		APROVADO	APROVADO	
	Campo vazio		[]		APROVADO	APROVADO	
Data da entrega	Data atual		06/08/2024		REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-116
	Data futura		15/08/2024		APROVADO	APROVADO	
	Data passada		01/08/2024		REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-118
	Campo vazio				REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-117
	Campo obrigatório		~~15/08/2024~~		APROVADO	APROVADO	
	Digitar manualmente		123456		APROVADO	APROVADO	
Período de locação	Um dia		Um dia		APROVADO	APROVADO	
	Dois dias		Dois dias		APROVADO	APROVADO	
	Três dias		Três dias		APROVADO	APROVADO	
	Quatro dias		Quatro dias		APROVADO	APROVADO	
	Cinco dias		Cinco dias		APROVADO	APROVADO	
	Seis dias		Seis dias		APROVADO	APROVADO	
	Sete dias		Sete dias		APROVADO	APROVADO	
Cor	Cinza		Cinza		APROVADO	APROVADO	
	Preto		Preto		APROVADO	APROVADO	
	Ambos		Ambos		APROVADO	APROVADO	
	Campo vazio		[]		APROVADO	APROVADO	
Comentário	O comprimento deve ser de 0 a 24 caracteres.	0,24	elainemateussss (15)	e (1) elainemateussssssssssss (23) elainemateussssssssssss (24) elainemateussssssssssss (25)	APROVADO	APROVADO	
	Maior que 24 caracteres	25	elainemateussssssssssssss (27)	elainemateussssssssssss (24) elainemateussssssssssss (25) elainemateussssssssssss (26)	REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-119 https://elainemtaraujo.atlassian.net/browse/BR-120 https://elainemtaraujo.atlassian.net/browse/BR-121
	Letras latinas		elainemateussss		APROVADO	APROVADO	
	Letras não latinas		정국		APROVADO	APROVADO	

	Hífen		elaine-		APROVADO	APROVADO	
	Espaço		elaine mateus		APROVADO	APROVADO	
	Números		elaine123		APROVADO	APROVADO	
	Barras		elaine/				
	Pontos		elaine.				
	Vírgulas		elaine,				
	Caracteres especiais		elaine@		REPROVADO	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-122
	Campo vazio		[]		APROVADO	APROVADO	

Descrição do teste	Etapas	Dados do teste (corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
Criar um entregador - POST /api/v1/courier						
Criar um entregador	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "saske".	{ "login": "ninja", "password": "1234", "firstName": "saske" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "login" com 1 caracter.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "n", o "password" = "1234" e o "firstName" = "saske".	{ "login": "n", "password": "1234", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-123
Enviar requisição POST com o campo "login" com 2 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ni", o "password" = "1234" e o "firstName" = "saske".	{ "login": "ni", "password": "1234", "firstName": "saske" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "login" com 3 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "nin", o "password" = "1234" e o "firstName" = "saske".	{ "login": "nin", "password": "1234", "firstName": "saske" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "login" com 9 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninjaaaaa", o "password" = "1234" e o "firstName" = "saske".	{ "login": "ninjaaaaa", "password": "1234", "firstName": "saske" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "login" com 10 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninjaaaaaa", o "password" = "1234" e o "firstName" = "saske".	{ "login": "ninjaaaaaa", "password": "1234", "firstName": "saske" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "login" com 11 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninjaaaaaaa", o "password" = "1234" e o "firstName" = "saske".	{ "login": "ninjaaaaaaa", "password": "1234", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-124
Enviar requisição POST com o campo "login" preenchido com números.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "12345", o "password" = "1234" e o "firstName" = "saske".	{ "login": "12345", "password": "1234", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-125
Enviar requisição POST com o campo "login" preenchido com caracteres especiais.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja@", o "password" = "1234" e o "firstName" = "saske".	{ "login": "ninja@", "password": "1234", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-126
Enviar requisição POST com o campo "login" preenchido com letras não latinas.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "사랑", o "password" = "1234" e o "firstName" = "saske".	{ "login": "사랑", "password": "1234", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-127
Enviar requisição POST com o campo "login" preenchido com hífen.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja-", o "password" = "1234" e o "firstName" = "saske".	{ "login": "ninja-", "password": "1234", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-128
Enviar requisição POST com o campo "login" preenchido com espaço.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "nin ja", o "password" = "1234" e o "firstName" = "saske".	{ "login": "nin ja", "password": "1234", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-129
Enviar requisição POST com o campo "login" preenchido com campo vazio.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "", o "password" = "1234" e o "firstName" = "saske".	{ "login": "", "password": "1234", "firstName": "saske" }	400 Bad request	400 Bad request	APROVADO	
Enviar requisição POST com o campo "password" com 3 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "123" e o "firstName" = "saske".	{ "login": "ninja", "password": "123", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-130
Enviar requisição POST com o campo '"password" com 4 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "saske".	{ "login": "ninja", "password": "1234", "firstName": "saske" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "password" com 5 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "12345" e o "firstName" = "saske".	{ "login": "ninja", "password": "12345", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-131
Enviar requisição POST com o campo "password" preenchido com uma string.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "amarelo" e o "firstName" = "saske".	{ "login": "ninja", "password": "amarelo", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-132
Enviar requisição POST com o campo "password" preenchido com caracteres especiais.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "123@" e o "firstName" = "saske".	{ "login": "ninja", "password": "123@", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-133
Enviar requisição POST com o campo "password" preenchido com letras não latinas.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "사랑" e o "firstName" = "saske".	{ "login": "ninja", "password": "사랑", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-133
Enviar requisição POST com o campo "password" preenchido com hífen.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "123-" e o "firstName" = "saske".	{ "login": "ninja", "password": "123-", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-135
Enviar requisição POST com o campo "password" preenchido com espaço.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "12 34" e o "firstName" = "saske".	{ "login": "ninja", "password": "12 34", "firstName": "saske" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-136
Enviar requisição POST com o campo "password" preenchido com campo vazio.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "" e o "firstName" = "saske".	{ "login": "ninja", "password": "", "firstName": "saske" }	400 Bad request	400 Bad request	APROVADO	
Enviar requisição POST com o campo "firstName" com 1 caracter.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "s".	{ "login": "ninja", "password": "1234", "firstName": "s" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-137
Enviar requisição POST com o campo "firstName" com 2 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "sa".	{ "login": "ninja", "password": "1234", "firstName": "sa" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "firstName" com 3 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "sas".	{ "login": "ninja", "password": "1234", "firstName": "sas" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "firstName" com 9 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "saskeeeee".	{ "login": "ninja", "password": "1234", "firstName": "saskeeeee" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "firstName" com 10 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "saskeeeeee".	{ "login": "ninja", "password": "1234", "firstName": "saskeeeeee" }	201 Created	201 Created	APROVADO	
Enviar requisição POST com o campo "firstName" com 11 caracteres.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "saskeeeeeee".	{ "login": "ninja", "password": "1234", "firstName": "saskeeeeeee" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-138
Enviar requisição POST com o campo "firstName" preenchido com números.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "12345".	{ "login": "ninja", "password": "1234", "firstName": "12345" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-139
Enviar requisição POST com o campo "firstName" preenchido com caracteres especiais.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "saske@".	{ "login": "ninja", "password": "1234", "firstName": "saske@" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-140
Enviar requisição POST com o campo "firstName" preenchido com letras não latinas.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "사랑".	{ "login": "ninja", "password": "1234", "firstName": "사랑" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-141
Enviar requisição POST com o campo "firstName" preenchido com hífen.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "saske-".	{ "login": "ninja", "password": "1234", "firstName": "saske-" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-142
Enviar requisição POST com o campo "firstName" preenchido com espaço.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "sas ke".	{ "login": "ninja", "password": "1234", "firstName": "sas ke" }	400 Bad request	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-143
Enviar requisição POST com o campo "firstName" preenchido com campo vazio.	1. Envie uma solicitação POST para /api/v1/courier com o "login" = "ninja", o "password" = "1234" e o "firstName" = "".	{ "login": "ninja", "password": "1234", "firstName": "" }	201 Created	201 Created	APROVADO	
Deletar um entregador - DELETE /api/v1/courrier/:id						
Deletar um entregador sem pedido associado	1. Envie uma solicitação DELETE para /api/v1/courrier/:id com o id do entregador.	Para id=1 /api/vi/courrier/1	200 OK	200 OK	APROVADO	
Deletar um entregador com pedido associado	1. Aceitar o pedido pela loja e associar o entregador. 2. Envie uma solicitação DELETE para /api/v1/courrier/:id com o id do entregador.	Para id=1 /api/vi/courrier/1	200 OK	404 Not Found	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-147
Deletar um entregador com ID inexistente	1. Envie uma solicitação DELETE para /api/v1/courrier/:id com o id inexistente.	Para id "88" (preencher com um id inexistente) /api/vi/courrier/88	404 Not Found	404 Not Found	APROVADO	
Deletar um entregador com o ID vazio	1. Envie uma solicitação DELETE para /api/v1/courrier/:id com o id vazio.	/api/vi/courrier/ { "id": "" }	400 Bad request	404 Not Found	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-144
Consultar um pedido - GET /api/v1/orders/track?t=track						
Consultar um pedido existente	1. Envie uma solicitação GET para /api/v1/orders/track?t=track com um número de pedido existente.	Para track=1234, fica /api/v1/orders/track?t=1234	200 OK	200 OK	APROVADO	
Consultar um pedido inexistente	1. Envie uma solicitação GET para /api/v1/orders/track?t=track com um número de pedido inexistente.	Para track "8999" (preencher com um track inexistente) /api/v1/orders/track?t=8999	404 Not Found	404 Not Found	APROVADO	
Consultar um pedido com o parâmetro "track" vazio.	1. Envie uma solicitação GET para /api/v1/orders/track?t=track com o parâmetro "track" vazio.	/api/v1/orders/track?t=	400 Bad request	400 Bad request	APROVADO	
Cancelar um pedido - PUT /api/v1/orders/cancel						
Cancelar um pedido existente não aceito pela loja	1. Envie uma solicitação PUT para /api/v1/orders/cancel com um número de pedido existente e não aceito pela loja.	Para track=1234, fica { "track": 1234 }	200 OK	400 Bad request	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-145
Cancelar um pedido existente aceito pela loja	1. Aceitar o pedido pela loja e associar o entregador. 2. Envie uma solicitação PUT para /api/v1/orders/cancel com um número de pedido existente e aceito pela loja.	Para track=1234, fica { "track": 1234 }	409 Conflict	400 Bad request	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-148
Cancelar um pedido inexistente	1. Envie uma solicitação PUT para /api/v1/orders/cancel com um número de pedido inexistente.	Para track 8999 (preencher com um track inexistente) { "track": 8999 }	404 Not Found	400 Bad request	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-146
Cancelar um pedido com o parâmetro "track" vazio.	1. Envie uma solicitação PUT para /api/v1/orders/cancel com um número de pedido inexistente.	{ "track": "" }	400 Bad request	400 Bad request	APROVADO	