VTEX

Accelerate
Commerce
Transformation

Introduzindo o VTEX IO

VTEX BOOK

VTEX

Accelerate
Commerce
Transformation

Quarta Edição - (Fevereiro 2020)

Comentários ao nosso endereço:

VTEX
Avenida Brigadeiro Faria Lima Ave, 4.440, 10 andar
Itaim Bibi, São Paulo, Brazil, 04538-132

Ao enviar informações à VTEX, você concede à VTEX o direito não exclusivo de usar ou distribuir as informações da maneira que julgar apropriada, sem incorrer em nenhuma obrigação para com você.

Sumário

Prefácio 08

1 Introdução 09

1.1 O que é VTEX IO 10

1.2 O que é REACT 11

1.3 O que significa programar REACT dentro d VTEX 11

1.4 CMS+VTEX IO 12

1.5 Benefícios do VTEX IO 12

2 Planejamento e implantação de projeto comVTEX IO 13

2.1 Performance e Usabilidade 14

2.2 Implantação 14

3 Desenvolvimento 15

3.1 Workspace 16

3.2 O que é um workspace? 17

3.3 Como criar um workspace? 17

4 Store Theme 20

4.1 O que é store theme? 21

4.2 Vinculando o store theme com seu diretório 21

4.3 Estrutura de pastas 23

4.4 Acessano seu ambiente de desenvolvimento 26

4.5 Templates 26

4.6 Estrutura em blocos 27

4.7 Criando um novo bloco 28

4.8 Customizando seu template 29

4.9 Estilo 29

4.10 Estilo Personalizado 30

Sumário

4.11 Fonte personalizada 32

4.12 Ícones personalizados 32

4.13 Flex layout 33

4.14 Outros componentes 34

4.15 Páginas Customizadas: como criar 34

5 Apps 36

5.1 Ambiente Enterprise x Ambiente Standart 37

5.2 O que é um app? 37

5.3 Linkando um app em sua loja 37

5.4 Criando e validando um app 38

6 Graphql 39

6.1 O que é o Graphql 40

6.2 Trabalhando com GraphQl em sua loja 40

6.3 Publicando sua loja 42

7 Administração 43

7.1 Site editor 44

7.2 Como utilizar o site editor? 44

7.3 Vinculando URL's em banners 45

7.4 Recomendações 48

8 Case: TEXAS BOOT 49

Texas Boot 50

Sobre a marca 50

8.1 Desafios 50

8.2 Resultados 51

Índice Imagens

Imagem 1 - Estrutura básica da loja 16
Imagem 2 - Instalação do VTEX CLI 17
Imagem 3 - Login 18
Imagem 4 - Login 18
Imagem 5 - Criação workspace 19
Imagem 6 - Store Theme 21
Imagem 7 - Preenchendo dados Store Theme 21
Imagem 8 - Compilando o tema 22
Imagem 9 - verificando/instalando 22
Imagem 10 - Estrutura de pastas 23
Imagem 11 - manifest.json 24
Imagem 12 - Pasta Store 24
Imagem 13 - Pasta Store 25
Imagem 14 - Pasta Styles 25
Imagem 15 - Acessando workspace criado 26
Imagem 16 - Estrutura em blocos 27
Imagem 17 - Lista de permissão 28
Imagem 18 - Novo bloco 28
Imagem 19 - Exemplo de dependência 28
Imagem 20 - Alterações refletidas automaticamente 29
Imagem 21 - Styles 30
Imagem 22 - Estilos personalizados 31
Imagem 23 - Estilo renderizado 31
Imagem 24 - Fontes personalizadas 32

Índice Imagens

Imagem 25 - Fontes personalizadas 32
Imagem 26 - Ícones personalizados 33
Imagem 27 - Flex Layout 33
Imagem 28 - Flex Layout 34
Imagem 29 - Criação página customizada 35
Imagem 30 - Criação página customizada 35
Imagem 31 - Linkando um app 38
Imagem 32 - GraphQl 40
Imagem 33 - Apollo 41
Imagem 34 - Buscando dados VTEX 41
Imagem 35 - Resultados da busca 42
Imagem 36 - Site Editor 44
Imagem 37 - Blocks 45
Imagem 38 - URL's em banner 46
Imagem 39 - URL's em banner 46
Imagem 40 - URL's em banner 47
Imagem 41 - URL's em banner 47
Imagem 42 - URL's em banner 48

VTEXbook CREATORS

Quarta edição - 2020

Todas as edições deste VTEXBOOK foram produzidas por especialistas de todo mundo.

O orientador deste projeto foi:

João N. de
A. Teixeira Neto
VTEX Brasil

Os autores da quarta edição foram:

Giovana Pereira
Maeztra

Rogério Nogueira
Maeztra

Agradecimentos pelos inestimáveis conselhos e orientações fornecidos na produção deste documento

Érick Herrera
Maeztra

Prefácio

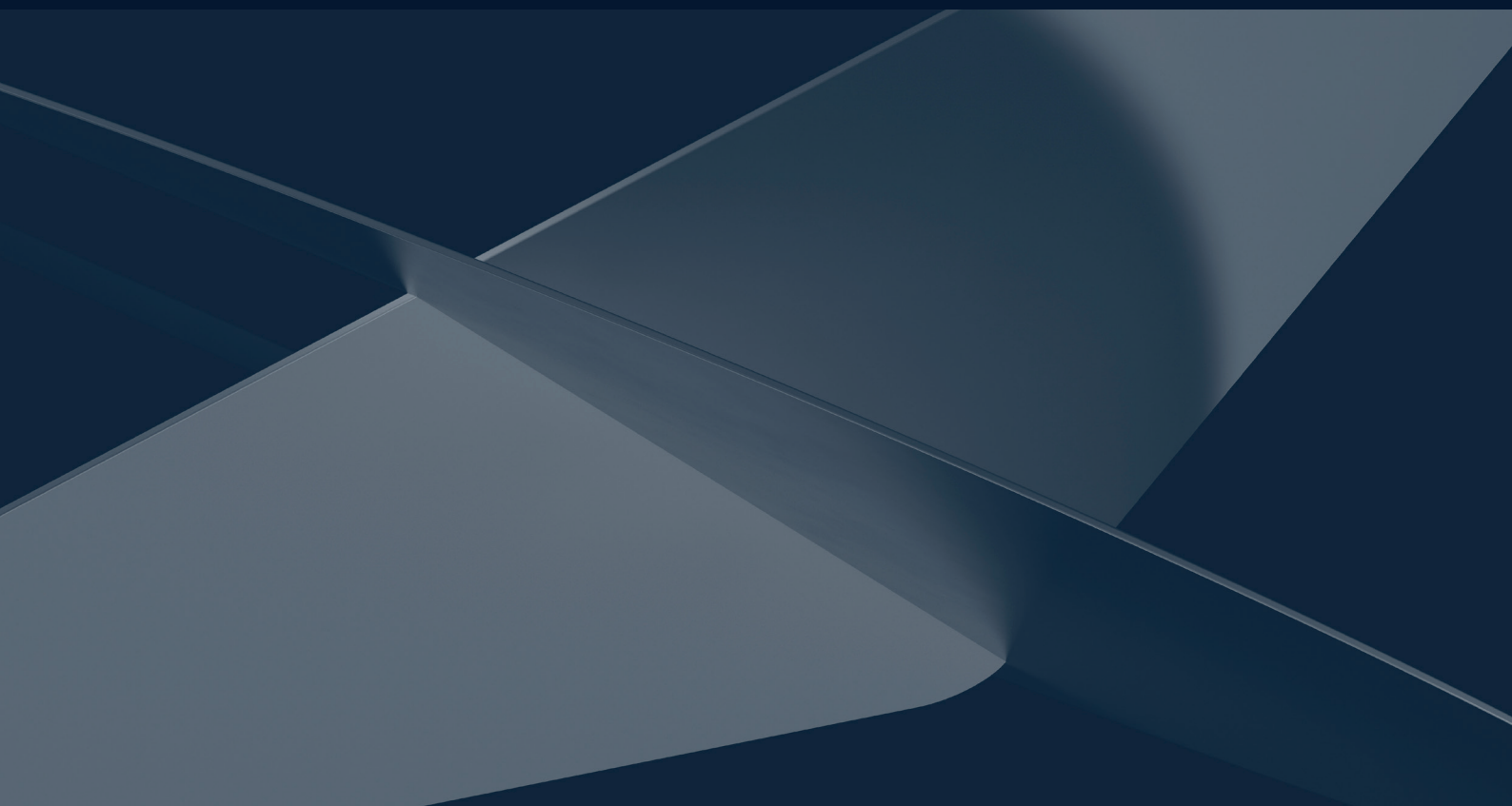
Com o objetivo de apresentar em linhas gerais algumas funcionalidades pertinentes dentro da usabilidade VTEX, esse VTEXBOOK pretende dar as primeiras explicações sobre as características e o uso do VTEX IO com React (também conhecido como React.js).

Veremos com apoio de imagens as formas de utilização do VTEX IO com React e também conhecer um pouco mais de seu histórico, características e vantagens de uso.

Supõe-se que o leitor tenha um conhecimento básico da VTEX e esteja familiarizado com programação.

01

Introdução



1.1 O que é VTEX IO

Uma plataforma nativa capaz de ajudar a entrega de soluções de negócio com mais agilidade e segurança. Ela permite o foco na experiência do usuário ao invés da preocupação extrema de lojistas em pontos como o desenvolvimento, por exemplo.

A VTEX IO é testada e escalada automaticamente, evitando maiores problemas com atualizações, já que a interface da loja independe do back-end de desenvolvimento.

Para desenvolver lojas com essa plataforma, é necessário conhecimento nas tecnologias TypeScript, React, GraphQL, .Net e Node e linguagens mais utilizadas no meio de desenvolvimento. Essas criadas até mesmo para aumentar a performance, além de proporcionar fácil instalação e uso a todos os desenvolvedores com apenas um comando inline no terminal ou qualquer outro serviço disponível.
Todo desenvolvimento com VTEX IO é sincronizado automaticamente com os servidores, o que permite que nenhum código seja mantido localmente, facilitando as atualizações e evitando problemas como o cache.

Capaz de proporcionar diversos workspaces, o VTEX IO possibilita testes A/B, evita quebras e possíveis erros em sua loja e aumenta a segurança na experiência de usuário. Além disso, é capaz de acompanhar a evolução da loja, ajudando com cada teste, progressivamente, evitando problemas inesperados.

1.2 O que é REACT

Desenvolvida pela equipe do Facebook em meados de 2011, o React é uma biblioteca em Javascript capaz de flexibilizar a criação de interfaces para o usuário. Por ser uma linguagem declarativa, é extremamente eficiente e traz apoio ao desenvolvimento.

Sendo uma biblioteca, e não um framework, a linguagem permite maior flexibilidade no desenvolvimento por parte do desenvolvedor, não ficando preso ou limitado por parte daquele framework.

1.3 O que significa programar REACT dentro da VTEX?

O React agiliza e flexibiliza o desenvolvimento de lojas dentro da VTEX, atuando em módulos para que o lojista foque na experiência de seu usuário e a aprimore sem afetar em qualquer funcionalidade ou usabilidade. Assim, torna seu ambiente mais seguro.

Significa segurança, rapidez na entrega e na visualização de soluções para o negócio, possibilitando a atualização direta com os servidores da loja.

1.4 CMS+VTEX IO

Quais seriam as diferenças entre o VTEX IO e o CMS atual da VTEX? Primeiro, utilizando novas tecnologias como o React, a loja se torna muito mais segura e estável por conta da separação da área de desenvolvimento com as implementações de negócio da parte do lojista.

Também é possível administrar, e até mesmo personalizar, a loja com mais facilidade com aplicativos voltados a esse uso, além de aumentar a experiência de usuário e a performance da loja.

Além disso, uma das maiores vantagens em se programar IO é a rapidez com que as atualizações e melhorias feitas na loja podem ser vistas instantaneamente. Ou seja, não há cachê.

A separação da área de administração com a área de desenvolvimento, por sua vez, é um dos aspectos mais importantes, pois garante maior segurança para ambos os lados, além de garantir maior experiência para o usuário.

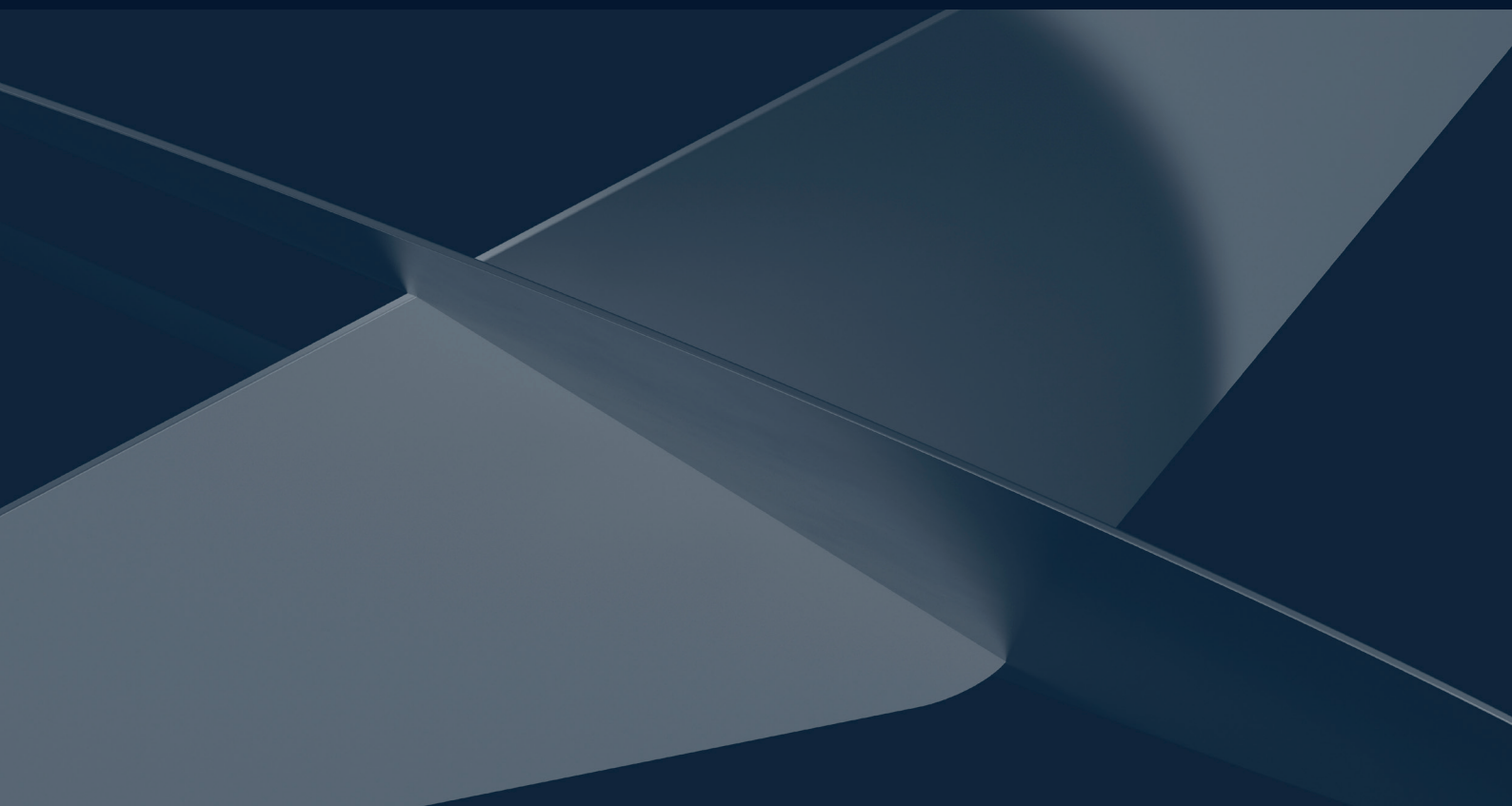
1.5 Benefícios do VTEX IO

As vantagens de trabalhar com VTEX IO são inúmeras, desde de agilidade, flexibilidade e foco na experiência do usuário, até segurança e estabilidade.

O desenvolvimento não atrapalha ou impede qualquer processo e solução de negócio, tendo dois ambientes separados e independentes. A liberdade e facilidade de ação são destacadas por aplicativos e outros frameworks utilizados que apoiam o desenvolvedor e o próprio lojista.

02

Planejamento e implantação de projeto com VTEX IO



2.1 Performance e Usabilidade

A usabilidade de uma loja desenvolvida em IO é focada para a experiência do usuário, de uma forma que ele não seja afetado negativamente por alterações e implementações novas que venham a ser inseridas na loja. Assim, toda transição é feita suavemente, de forma segura e estável.

Com a utilização das melhores bibliotecas e frameworks atuais, a loja desenvolvida em IO é modularizada e componentizada, trazendo um código mais limpo e organizado, o que possibilita o aumento de performance da loja.

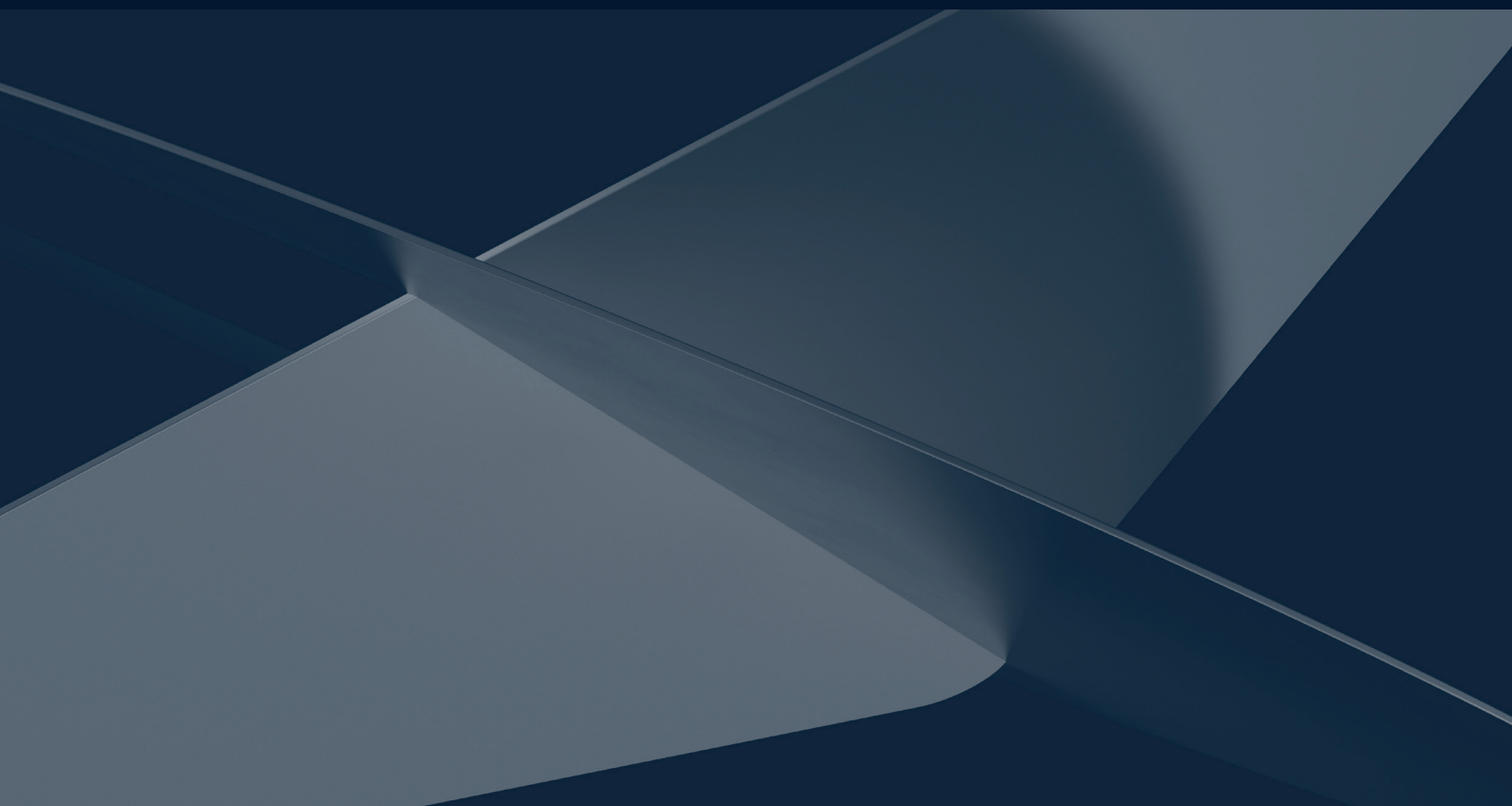
2.2 Implantação

Alguns passos são necessários para iniciar a implementação de uma loja em IO. É preciso preparar um ambiente para conversar com os servidores VTEX, além de um repositório capaz de manter uma conversa segura e estável, favorável a manutenção e atualização de suas versões desenvolvidas.

Iremos abordar, passo a passo, todas as necessidades e peculiaridades da plataforma, facilitando assim o desenvolvimento e manutenção de uma loja desenvolvida em VTEX IO.

03

Desenvolvimento



3.1 Workspace

O desenvolvimento de uma loja começa quando temos um workspace para trabalhar. Porém, para termos esse espaço, é preciso seguir uma série de configurações, como instalação de frameworks e aplicativos, tudo em um processo básico via comando online. Podemos ver no print abaixo, montado pela VTEX, a estrutura básica da loja em IO. Assim, o lojista precisa ter uma conta ativa para iniciar o projeto:

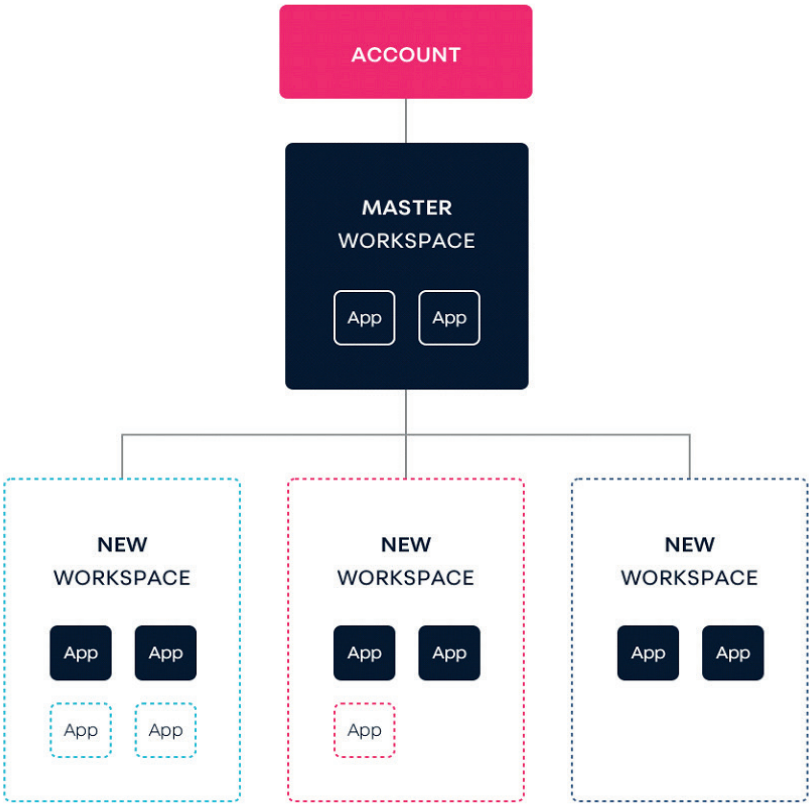


imagem 1 - Estrutura básica da loja

3.2 O que é um workspace?

É a área de trabalho para o desenvolvimento de sua loja, possuindo toda a estrutura e apresentação inicial, sem estilizações.

Como apontado no print acima, existe o master workspace, onde são vistas apenas as aplicações e estilizações já aprovadas e testadas. Elas aprimoram a experiência do usuário e, geralmente, sendo o indicado, não é um espaço para testes e trabalhos. Por meio deste master, é possível criar outros, iguais ou limpos, para iniciar o seu desenvolvimento de aplicações.

3.3 Como criar um workspace?

Antes de criarmos o workspace, é preciso:
Ter instalado em sua máquina o Node.JS e/ou Yarn.
Fazer a instalação do VTEX CLI em sua máquina com o seguinte comando (por meio do terminal de linha de comando):

`npm i -g vtex`
ou
`yarn global add vtex`

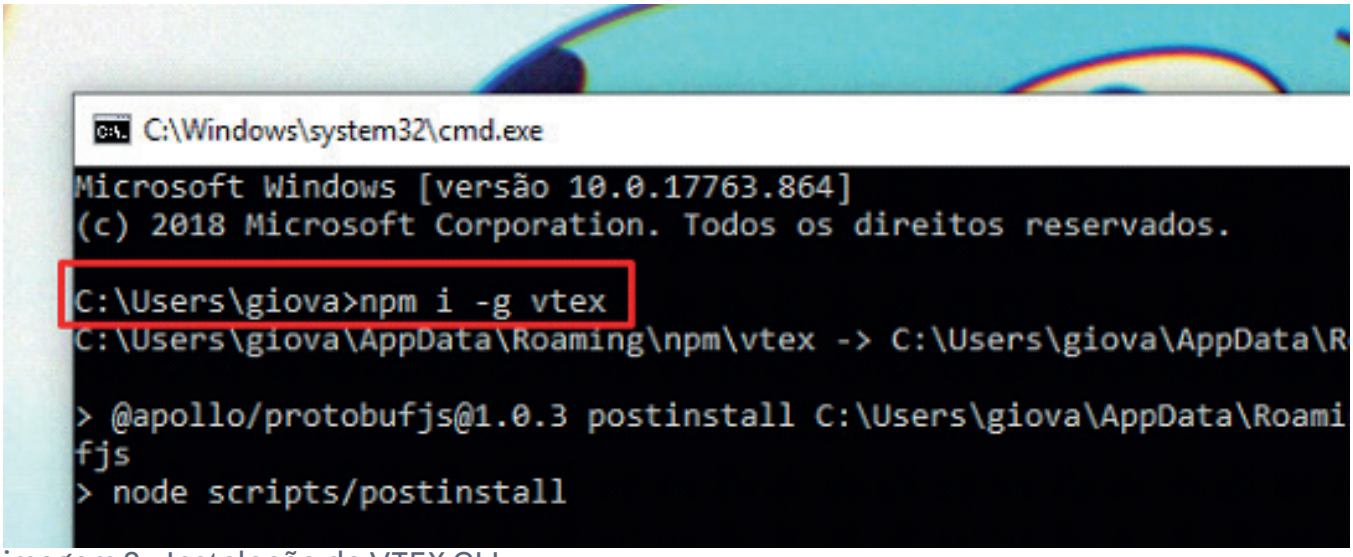


imagem 2 - Instalação do VTEX CLI

Logar na conta a ser trabalhada com o seguinte comando

```
vtex login -a {accountName}
```

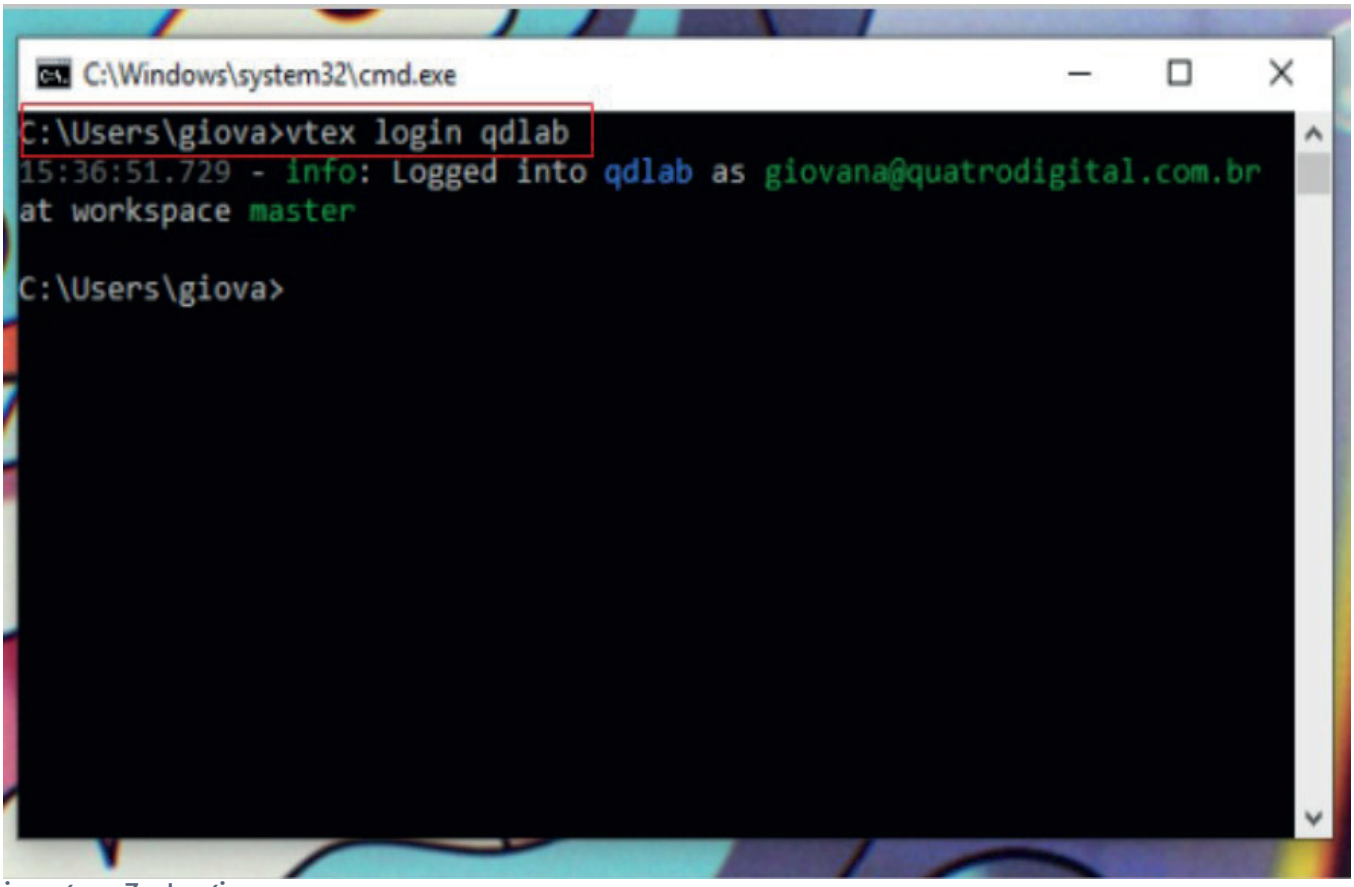


imagem 3 - Login

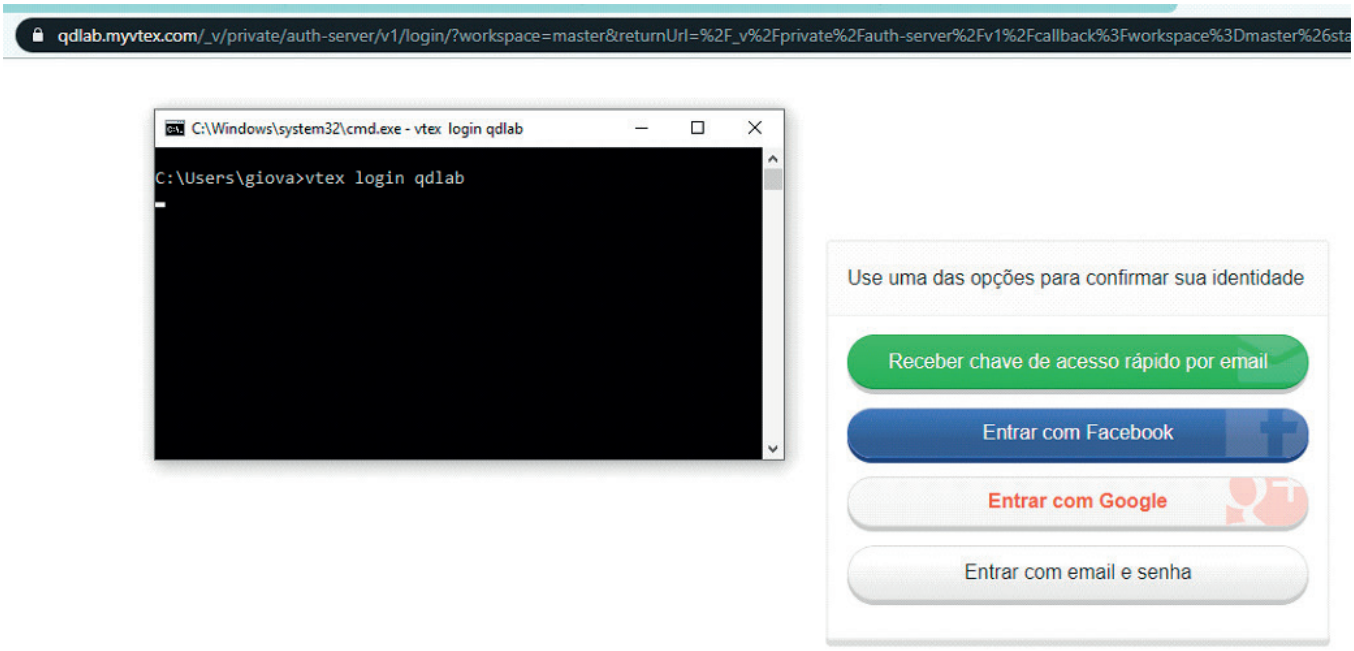


imagem 4 - Login

Depois de logar, você estará dentro do workspace master. Para desenvolver suas aplicações, é necessário criar outro workspace com o seguinte comando:
vtex use {workspaceName}.

Insira o nome que gostaria de dar a esse espaço (recomendamos ser o nome da loja com indicação de desenvolvimento).

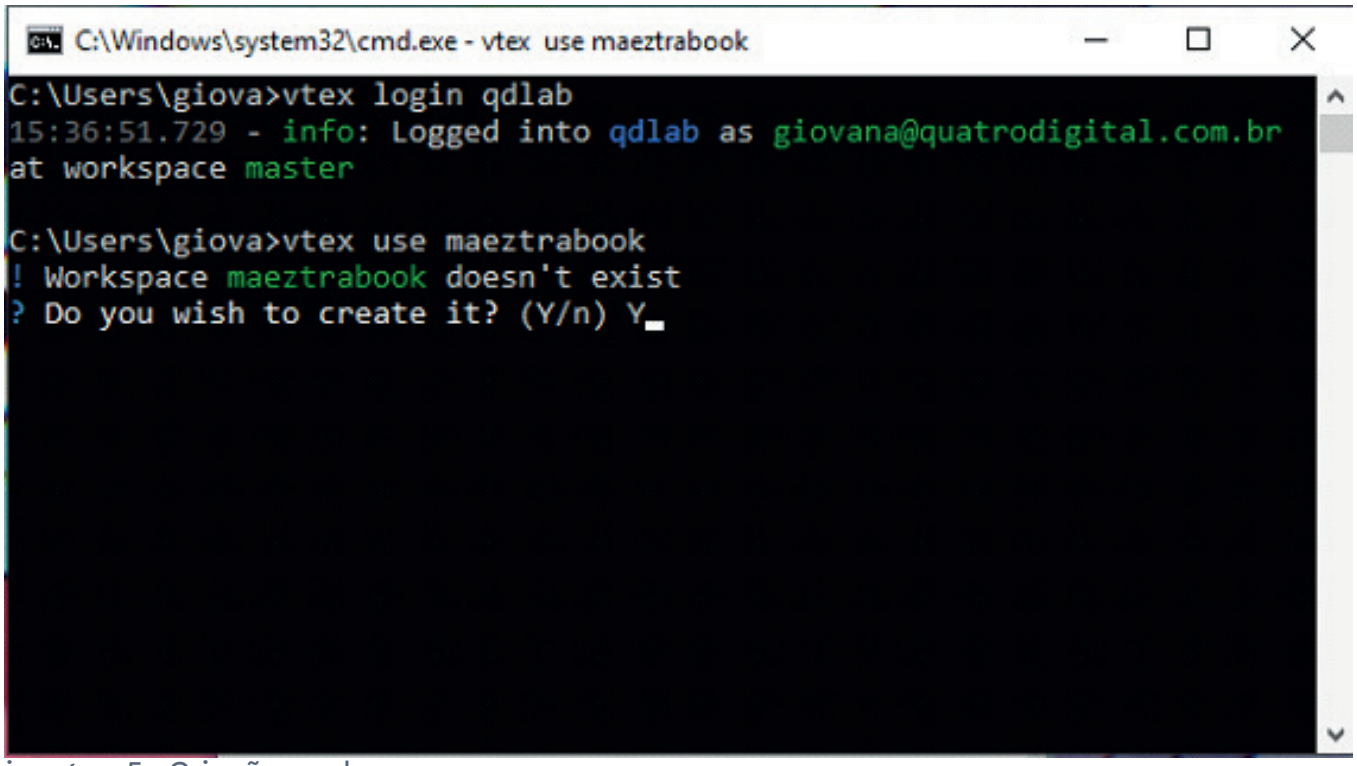
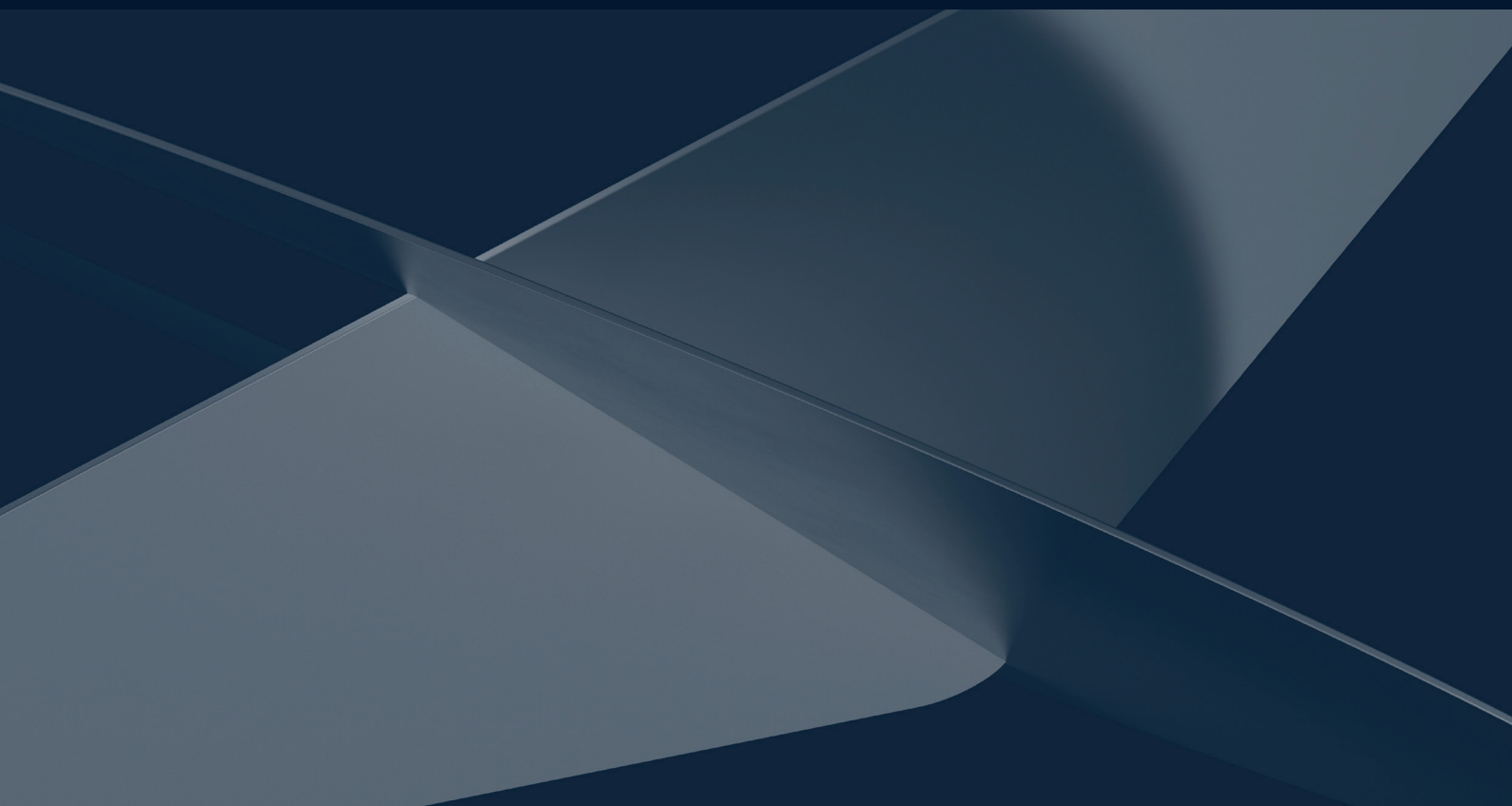


imagem 5 - Criação workspace

Acesse o workspace criado com o seguinte link:
https://{workspace}--{account}.myvtex.com
Substitua o {workspace} e o {account} pelas informações respectivas criadas anteriormente.
Ex.: https://qdlab--maeztrabook.myvtex.com/

04

Store Theme



4.1 O que é store theme?

É o tema padrão que pode ser instalado em toda loja recém configurada, com um estilo responsivo e básico, onde será possível declarar novos templates e estilos, além de criar novos componentes e alterar cores e tipografias.

4.2 Vinculando o store theme com seu diretório

1. Acesse o diretório desejado a ser baixado os arquivos iniciais usando o terminal de linha de comando.

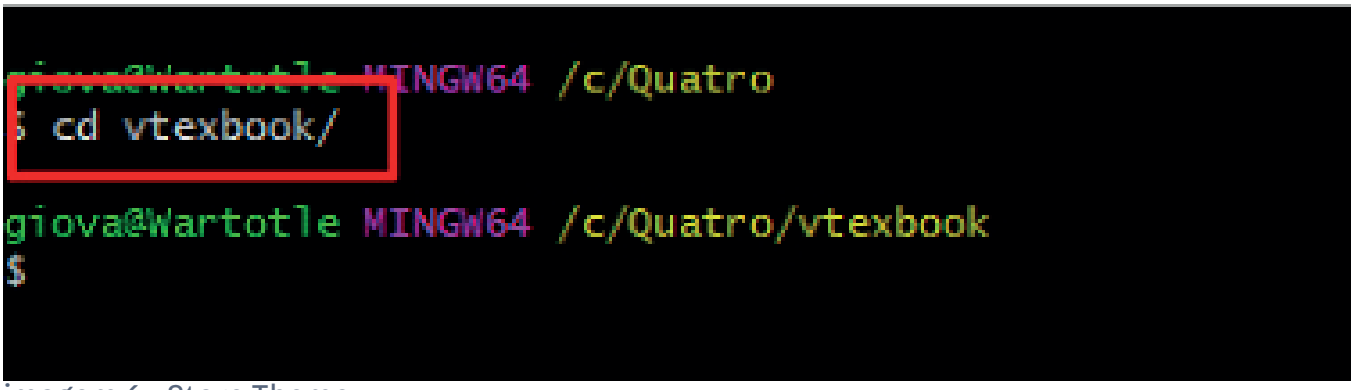


imagem 6 - Store Theme

2. Dentro da pasta, execute o comando vtex init e selecione a opção de store-theme. Siga com o preenchimento das informações, como vendor, name, title e description. OBS.: apenas a informação de vendor é obrigatória para continuar.

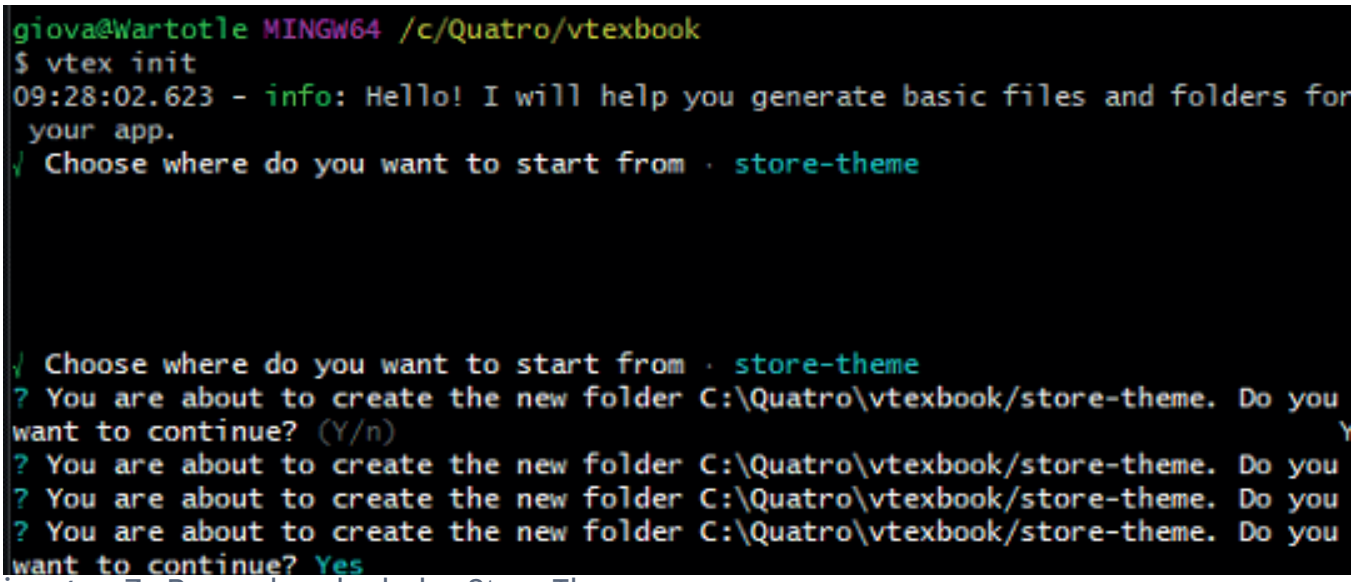


imagem 7 - Preenchendo dados Store Theme

3. Com a store theme copiada para o seu diretório, inicie em seguida o comando vtex link para compilar o tema e publicar no workspace logado anteriormente.

```
giova@Wartotle MINGW64 /c/Quatro/vtexbook
$ cd store-theme/

giova@Wartotle MINGW64 /c/Quatro/vtexbook/store-theme
$ vtex link
? Do you want to use the previous login details? (giovana@q
 qdlab / maeztrabook) (Y/n)
? Do you want to use the previous login details? (giovana@q
? Do you want to use the previous login details? (giovana@q
? Do you want to use the previous login details? (giovana@q
 qdlab / maeztrabook) Yes
09:35:42.867 - info: Logged into qdlab as giovana@quatrodig
ace maeztrabook
09:35:48.205 - info: Linking app qdlab.vtexbook@0.1.0
09:35:48.294 - info: Sending 28 files
09:35:50.536 - info: Build accepted for qdlab.vtexbook@0.1.
k vtex.builder-hub@0.183.2
```

imagem 8 - Compilando o tema

4. Para que você possa trabalhar perfeitamente, é preciso verificar/instalar algumas outras “dependências”, que seriam a vtex.store-sitemap, vtex.store e vtex.builder-hub, nessa sequência.

```
giova@Wartotle MINGW64 /c/Quatro/vtexbook/store-theme
$ vtex install vtex.store-sitemap@1.x
09:46:00.120 - info: Installed app vtex.store-sitemap@1.x successfully

giova@Wartotle MINGW64 /c/Quatro/vtexbook/store-theme
$ vtex install vtex.store@2.x
09:47:12.593 - info: Installed app vtex.store@2.x successfully

giova@Wartotle MINGW64 /c/Quatro/vtexbook/store-theme
$ vtex install vtex.builder-hub
09:47:49.586 - info: Installed app vtex.builder-hub successfully
```

imagem 9 - verificando/instalando

5. Inicie em seguida o comando `vtex link` para compilar o tema e publicar no workspace logado anteriormente.

Pronto! Agora podemos seguir com o desenvolvimento.

4.3 Estrutura de pastas

Ao abrir o editor de código de sua preferência, é possível encontrar a estrutura de pastas que foi baixada. Agora, iremos explorar as pastas e arquivos que serão mais utilizadas no desenvolvimento a fim de compreender sua estrutura total e como trabalhar dentro dela.
Observe a imagem abaixo.

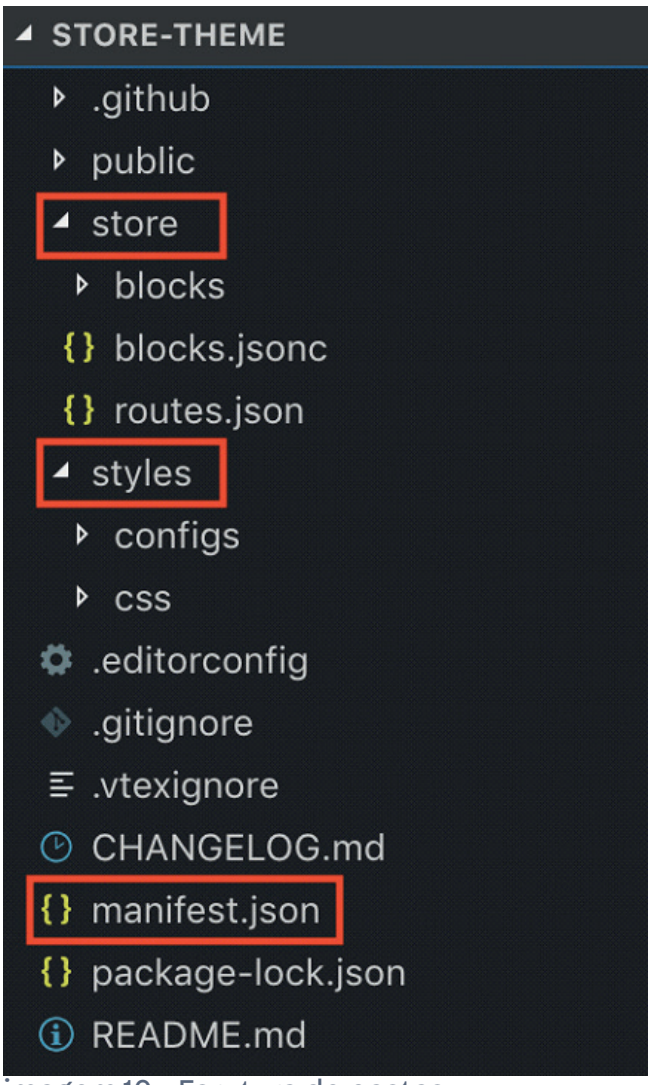


imagem 10 - Esrutura de pastas

O arquivo manifest.json carrega diversas informações importantes, inclusive das quais preenchemos as informações no ínicio da instalação do store theme, tais como vendor, name, description, etc.

Imagem 11 - manifest.json

A pasta store é o local onde poderá criar/ configurar os componentes de suas páginas e suas propriedades. Trata-se da estrutura da loja em si.

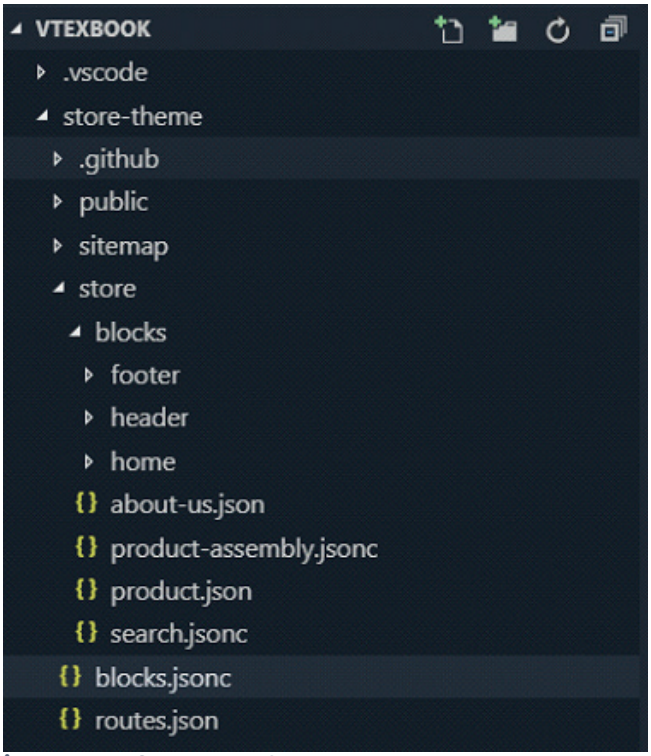


imagem 12 - Pasta Store

imagem 13 - Pasta Store

Para customizar e setar os estilos da loja, a pasta styles traz uma série de arquivos e possibilidades que serão exploradas mais para a frente neste livro.

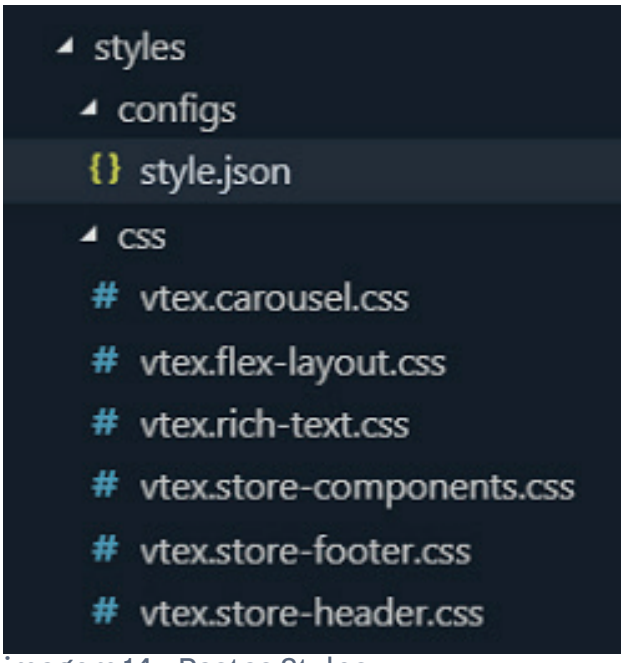


imagem 14 - Pastas Styles

4.4 Acessando seu ambiente de desenvolvimento

Para visualizar as alterações e instalações feitas, é preciso acessar o link de seu workspace com o vtex link rodando sempre. Isso garante que tudo seja compilado e enviado a ele.
Para acessar o workspace criado, basta utilizar o seguinte link:

https://{workspace}--{account}.myvtex.com
Substitua o {workspace} e o {account} pelas informações respectivas criadas anteriormente.
Ex.: https://qdlab--maeztrabook.myvtex.com/

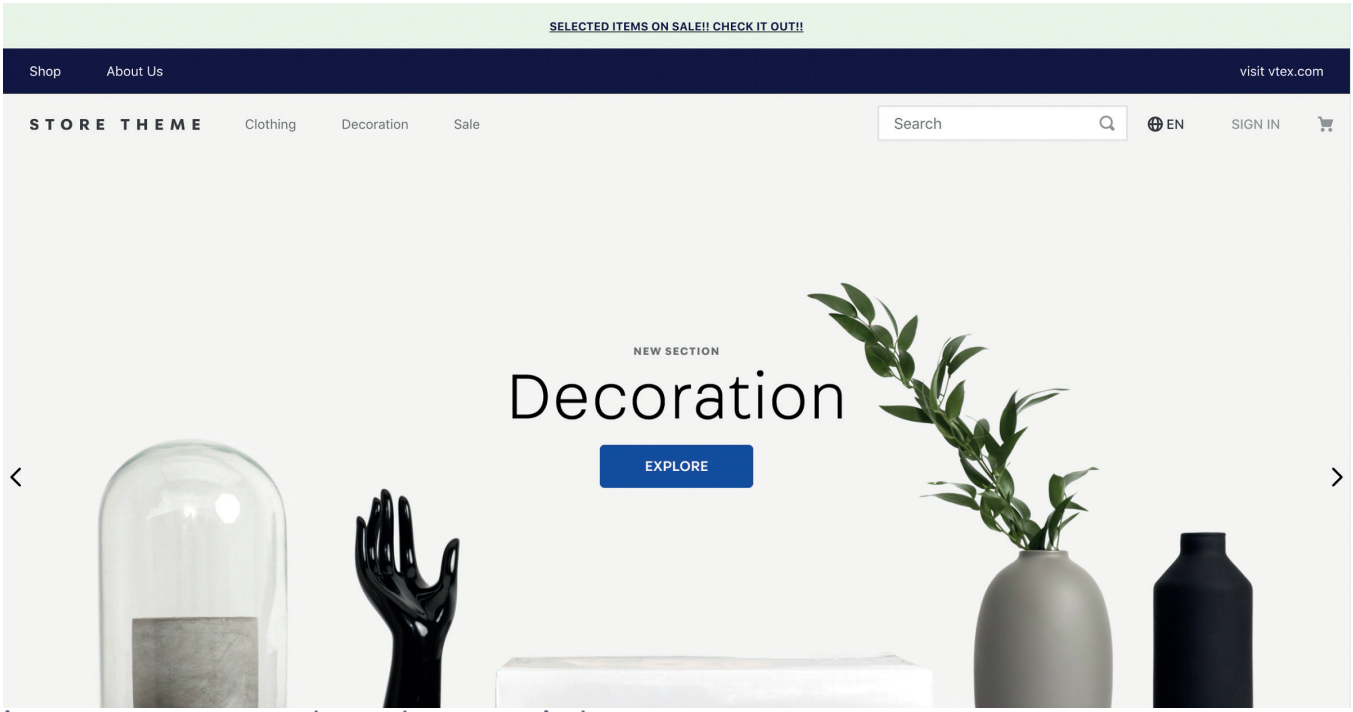


imagem 15 - Acessando workspace criado

4.5 Templates

Toda loja criada possui diversos templates disponíveis a serem configurados. Porém, para facilitar o desenvolvimento, a store them já traz templates prontos, tais como home, produto e busca.

O que resta a ser feito é configurar componentes, incluindo ou excluindo-os, usando o blocks.

4.6 Estrutura em blocos

Quando criamos um novo bloco, estamos disponibilizando uma área para inserir um novo componente em uma página, assim como podemos excluir um desses componentes em qualquer outro bloco.

A pasta responsável pela organização dos blocos é a store. Por padrão, existe o arquivo blocks.jsonc, mas você pode criar outros arquivos que desejar.

Utilize o editor de código de sua preferência e siga o caminho de pastas até encontrar o arquivo para entendermos como funcionam os blocos:

store > blocks> home> home.jsonc

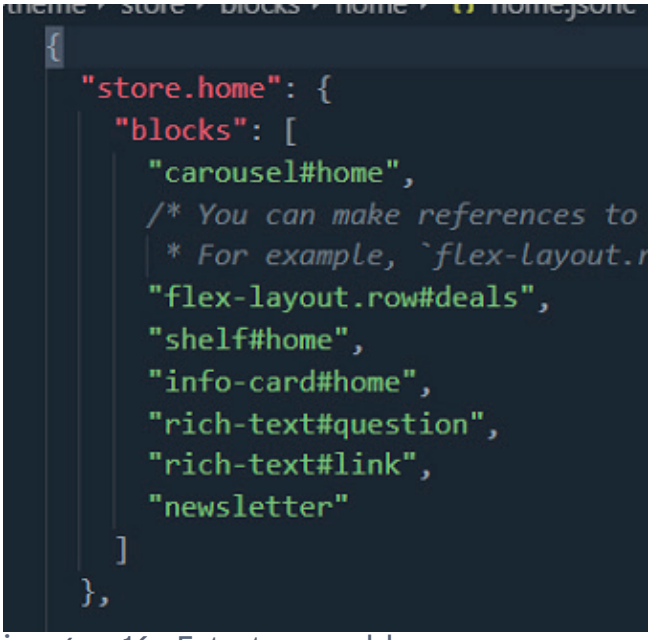


imagem 16 - Estrutura em blocos

No trecho de código store.home, estamos declarando sete blocos. Isso significa que a página home da loja possui essas sete informações exibidas (dependendo, obviamente, de seu conteúdo).

Repare que a ordem desses blocos é significativa, pois essas posições serão exibidas na tela para o usuário.

Cada um desses blocos deve ser declarado no mesmo arquivo, como no print abaixo. Eles podem ser customizados conforme suas dependências ou permissões de execução.

Você também pode conferir a lista de permissões dos blocos na documentação do VTEX IO, nesse link.

```
"rich-text#question": {
  "props": {
    "text": "**This is an example store built using the VTEX platform.\nWant to know more?**",
    "blockClass": "question"
  }
},
```

imagem 17 - Lista de permissão

4.7 Criando um novo bloco

Nesse mesmo arquivo, vamos adicionar um bloco de infocard. Para isso, precisamos nomear nosso novo bloco e, caso precise declarar a dependência, insira na frente do nome único.

```
"store.home": {
  "blocks": [
    "carousel#home",
    /* You can make references t
     * For example, `flex-layout
    "flex-layout.row#deals",
    "shelf#home",
    "info-card#sales"
    "info-card#home",
    "rich-text#question",
    "rich-text#link",
    "newsletter"
  ]
},
```

imagem 18 - Novo bloco

```
"info-card#sales": {
  "props": {
    "isFullModeStyle": false,
    "textPosition": "center",
    "imageUrl": "http://cybercitycomix.com/wp-content/uploads/2015/08/Sale-sign.jpg",
    "headline": "SALES",
    "callToActionText": "DISCOVER",
    "callToActionUrl": "/sale/d",
    "textAlignment": "center"
  }
},
```

imagem 19 - Exemplo de dependência

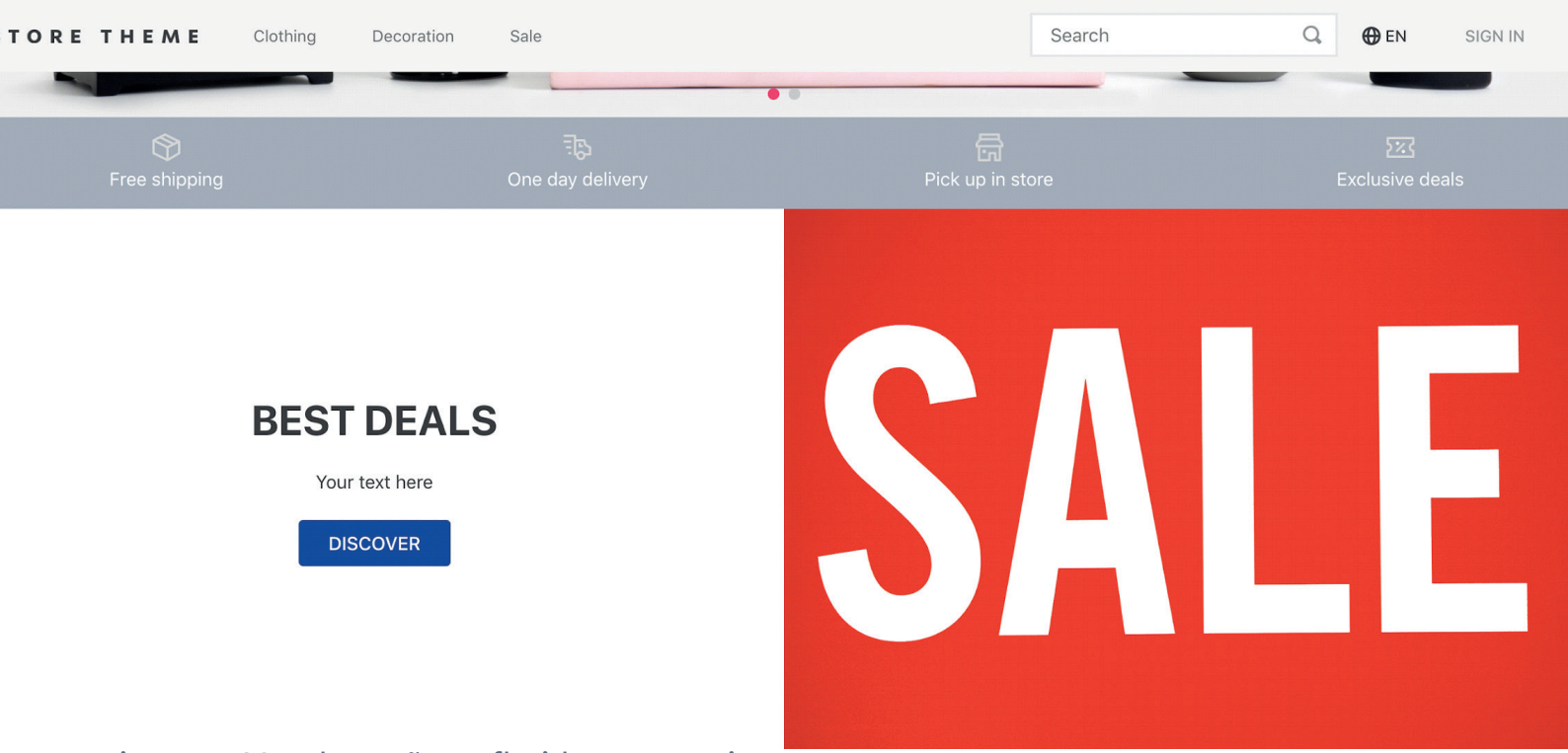


imagem 20 - Alterações refletidas automaticamente

Com o comando vtex link ativado, as alterações serão refletidas automaticamente no workspace em que estão sendo desenvolvidas. Para verificar, basta atualizar utilizando o link de seu workspace no navegador.

4.8 Customizando seu template

Agora que já vimos como criar novos blocos e inserir as informações desejadas, podemos personalizar cada bloco, dependendo das necessidades do lojista, por meio de estilos.

4.9 Estilo

É possível personalizar sua loja com as predefinições no arquivo padrão da store theme no arquivo style.json na pasta styles.

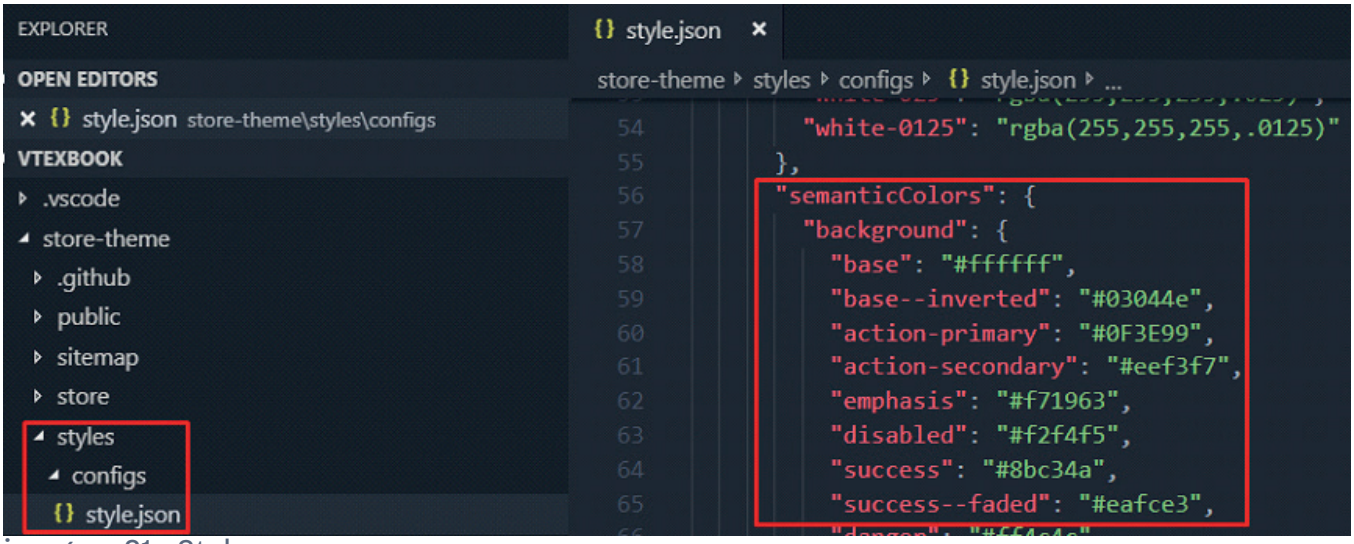


imagem 21 - Styles

A VTEX possui o VTEX Styleguide para ajudar a compreender cada uma dos itens dentro da estrutura deste arquivo, assim como facilitar o desenvolvimento e a personalização de cada item.

Existe também um bloco com as variáveis de cores padrões do site, onde você pode facilmente alterar de uma cor para a outra.

4.10 Estilo personalizado

Caso tenha necessidade de alterar o estilo da loja além do que é possível por meio das variáveis, é preciso seguir as seguintes etapas:

1. Crie um arquivo `css` dentro da pasta `styles/css` que tenha o nome do componente que deseja personalizar.

Ex.: Para alterar o header, é preciso criar o arquivo `vtex.store-header.css`, baseando-se na classe que aparece no HTML.

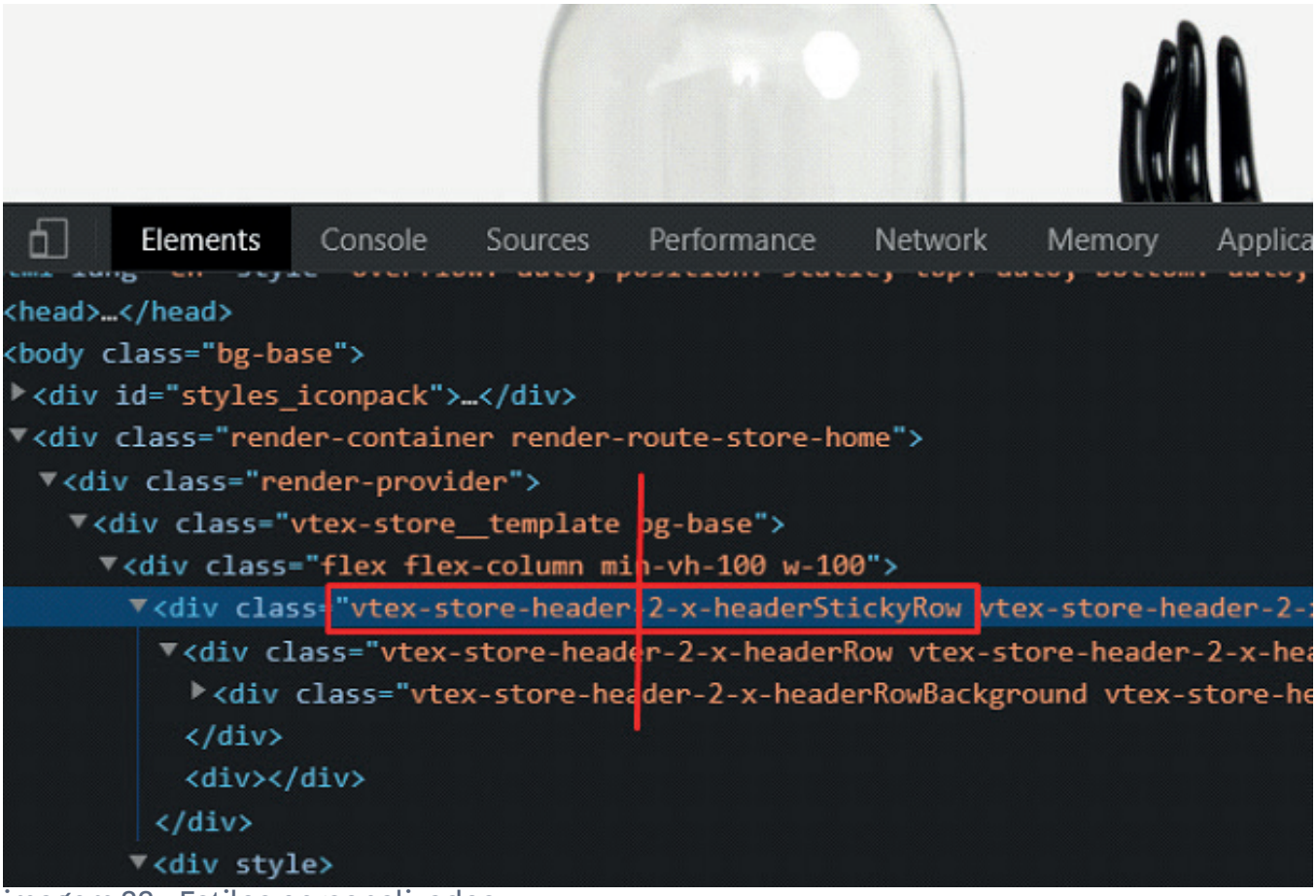


imagem 22 - Estilos personalizados

No print acima, é possível ver que, por meio de uma classe montado pelo IO, temos o início da classe sendo o nome do componente. O nome do arquivo, vtex.store-header.css , só sofre uma alteração do primeiro " - " para " . ".

Automaticamente , ao utilizar o vtex link, um arquivo geral será criado e seu estilo renderizado no componente por meio do navegador.

2. Após isso, todas as classes que começarem com vtex-store-header podem ser inseridas nesse arquivo e estilizadas. Não é preciso inserir o nome do componente dentro do arquivo (vtex-store-header-2-x- por exemplo), apenas a última parte.
Ex.:

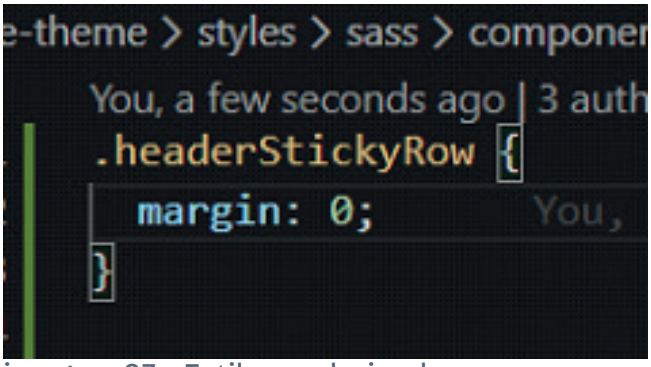


imagem 23 - Estilo renderizado

4.11 Fonte personalizada

Caso deseje alterar a fonte da loja, basta fazer a importação da nova. Recomenda-se utilizar o Google Fonts no arquivo vtex.store.css e usar o font-family nas classes necessárias.

```
Gabriel Leocadio, a month ago | 1 author (Gabriel Leocadio)
1  // Google Fonts - Noto Sans
2  @import url('https://fonts.googleapis.com/css?family=Noto+Sans&display=swap');
```

imagem 24 - Fontes Personalizadas

```
.vtex-shelf-1-x-title {
  font-family: "Noto Sans",sans-serif;
  font-weight: 600;
  font-size: 16px;
  line-height: 22px;
  text-transform: uppercase;
  color: #212121;
  text-align: left;
  display: block;
}
```

imagem 25 - Fontes personalizadas

4.12 Ícones personalizados

O VTEX IO utiliza seus ícones em SVG. Portanto, se deseja alterá-los, é preciso baixar o arquivo iconpack.svg (clique aqui para saber mais) dentro de uma pasta iconpacks, que, por sua vez, fica na pasta styles.
Procure pelo nome do ícone no “id” e altere o ícone para o que desejar.

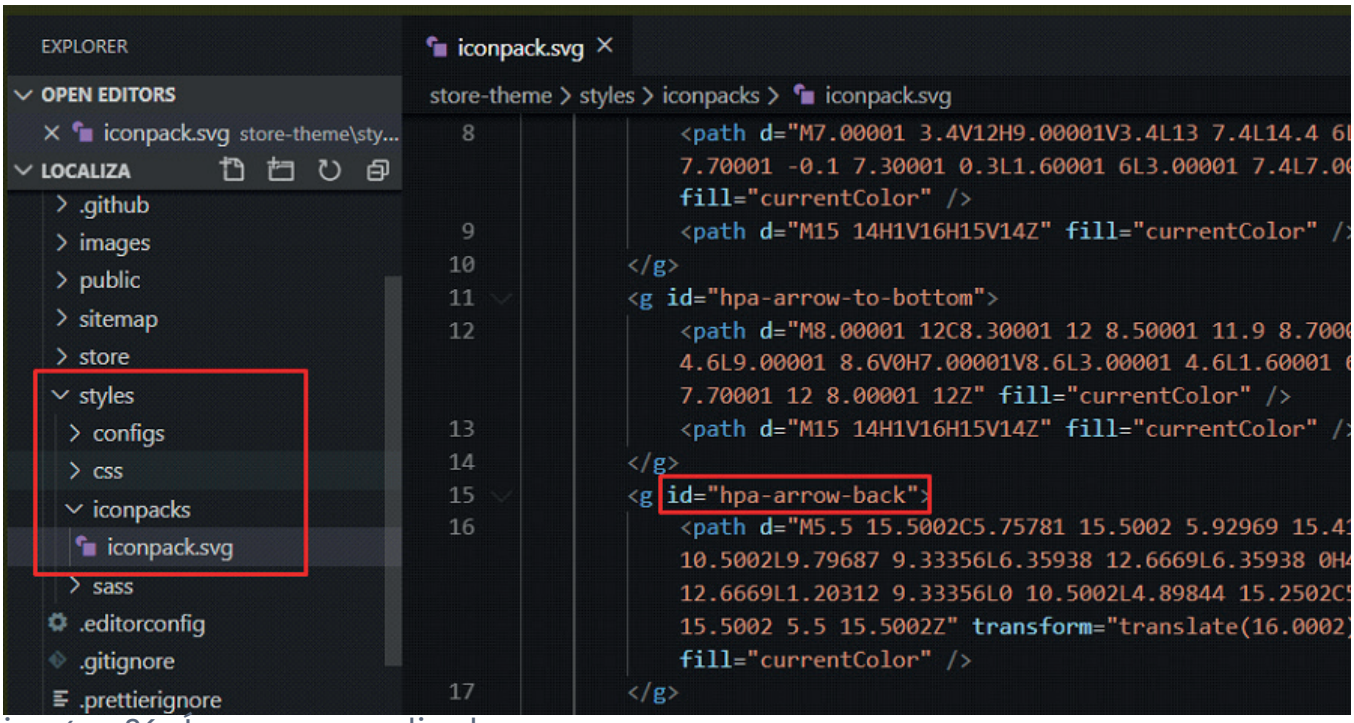


imagem 26 - Ícones personalizados

4.13 Flex layout

O flex layout é utilizado para estruturar a loja, criando colunas e linhas responsivas e utilizando os atributos flex do css. Com isso, facilita alinhamentos, por exemplo.
A utilização básica do flex layout é muito simples.

1.Primeiro, você deve criar uma linha no blocks chamando o flex-layout.row

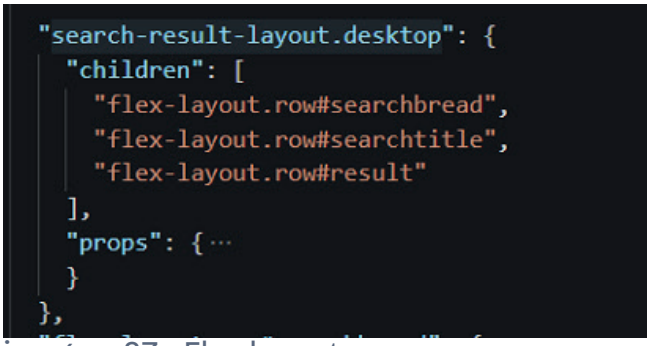


imagem 27 - Flex layout

2.Depois, utilize o children, ainda dentro do `blocks`, como `flex-layout.col`

```
"flex-layout.row#result": {
  "children": ["flex-layout.col#filter", "flex-layout.col#content"],
  "props": {
    "preventHorizontalStretch": true,
    "fullWidth": true
  }
},
"flex-layout.col#filter": {
  "children": ["filter-navigator.v3"],
  "props": {
    "blockClass": "filterCol"
  }
},
```

imagem 28 - Flex Layout

Para saber melhor todas as propriedades do Flex Layout e conhecer todas as possibilidades a serem trabalhadas em seu layout, acesse a documentação do VTEX IO linkada acima.

4.14 Outros componentes

Com o VTEX IO, também é possível estilizar muitos componentes, como criar uma vitrine de produtos em carroussel, por exemplo.

Colocamos aqui alguns exemplos que podem ajudar na construção básica de uma loja. Porém, caso queira conhecer outras possibilidades em mais detalhes, acesse a documentação do VTEX IO.

4.15 Páginas customizadas: como criar

Lojas possuem muitas páginas institucionais e, para comportar essa necessidade, o VTEX IO possibilita a criação de páginas institucionais novas.

Basta seguir o passo a passo abaixo:

1.Acesse o arquivo `routes.json` dentro de `store` e crie uma rota personalizada da seguinte forma:

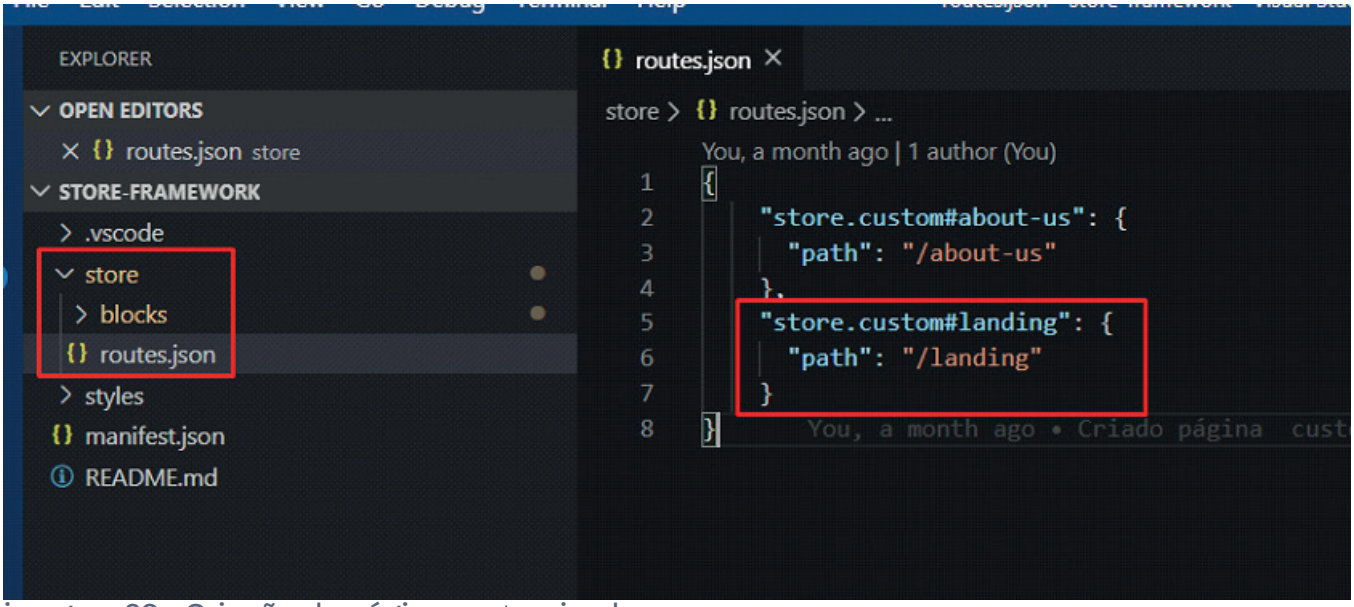


imagem 29 - Criação de página customizada

2.Após, crie um arquivo json com o nome de sua página (o que você colocou após a " # "), da seguinte forma:

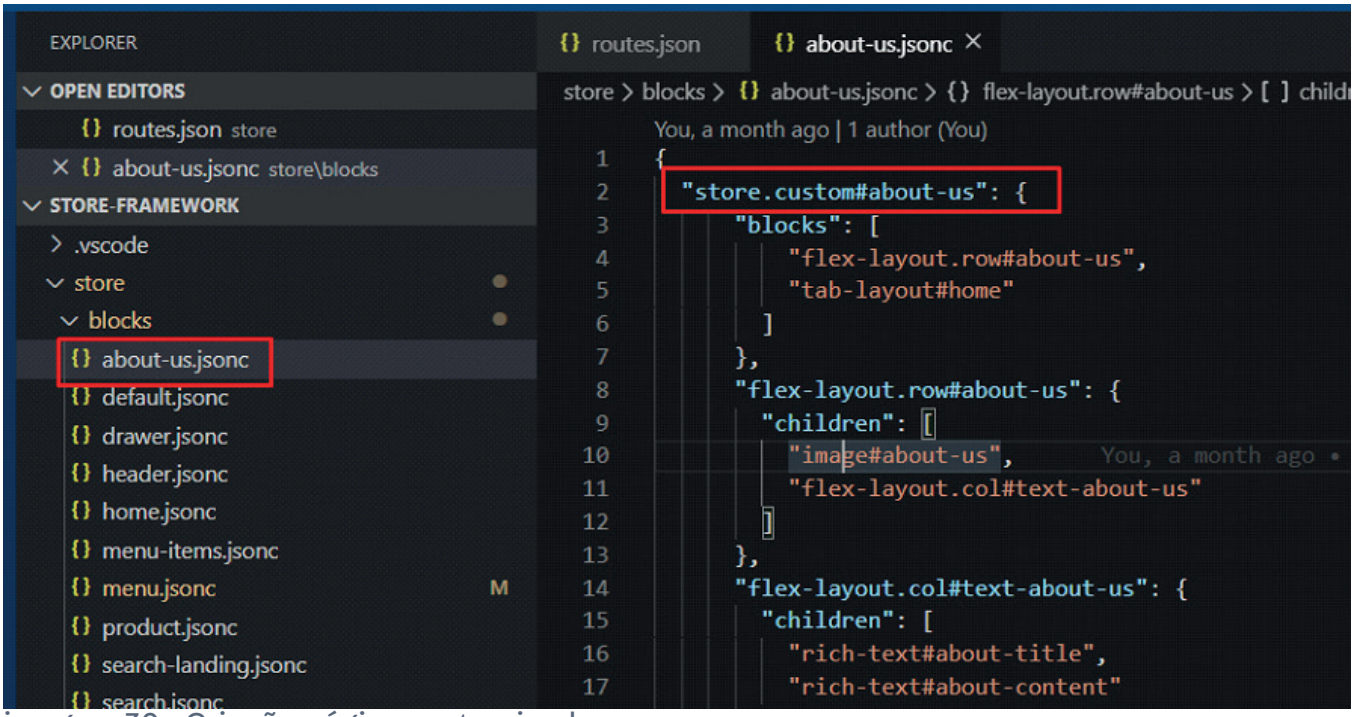
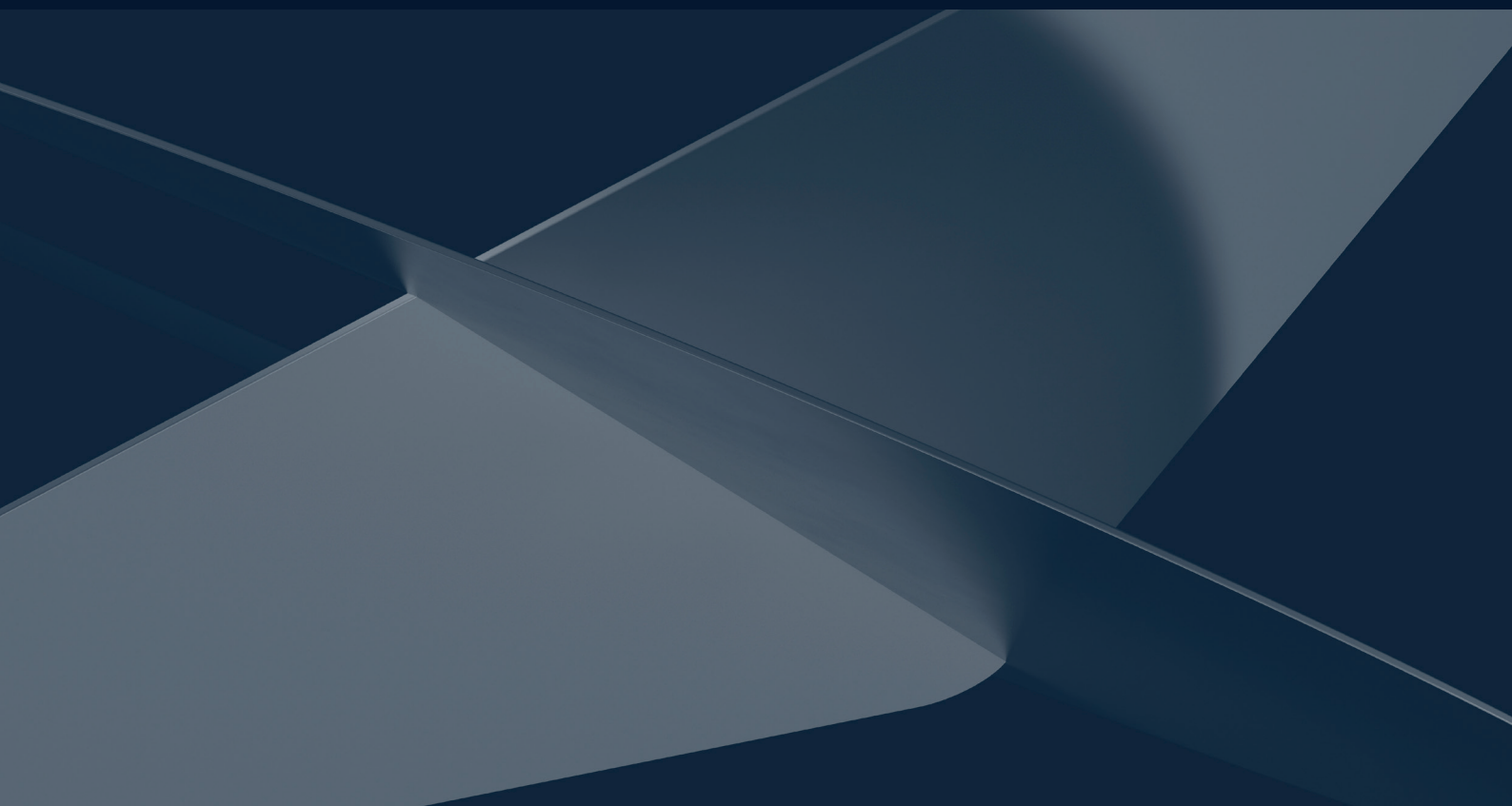


imagem 30 - Criação página customizada

3.Depois, basta acessar a página pela url criada dentro do arquivo routes e personalizar sua página utilizando os componentes do VTEX IO.

05

Apps



5.1 Ambiente Enterprise x Ambiente Standart

O VTEX IO possibilita desenvolver e instalar inúmeros apps (aplicativos) que podem agregar valor à sua loja.
É possível usar, criar e customizar apps, porém é preciso se atentar ao tipo de ambiente que a loja possui.

As permissões de utilização do VTEX IO, como estender componentes para apps personalizados, por exemplo, são diferenciadas para ambientes standart ou enterprise.

Um ambiente standart só permite a utilização dos componentes padrões do IO. Já o enterprise permite a customização de componentes.

5.2 O que é um app?

Diferente de componentes, sejam estendidos ou não, aplicativos dentro do VTEX IO atuam por si só. Há uma lógica dentro dele que é única para ser utilizada, como se fosse um conjunto de componentes que possuem uma sinergia.

5.3 Linkando um app em sua loja

Os apps são caracterizados, principalmente, por terem obrigatoriamente que serem incluídos no `manifest.json` do tema da sua loja.

Por exemplo, se deseja usar o app de integração com o Wordpress, o primeiro passo a ser seguido é instalar em sua loja pelo comando: `vtex i vtex.wordpress-integration@0.x -g`

Após rodar esse comando, é necessário inserir a linha com a versão dentro de seu `manifest.json`

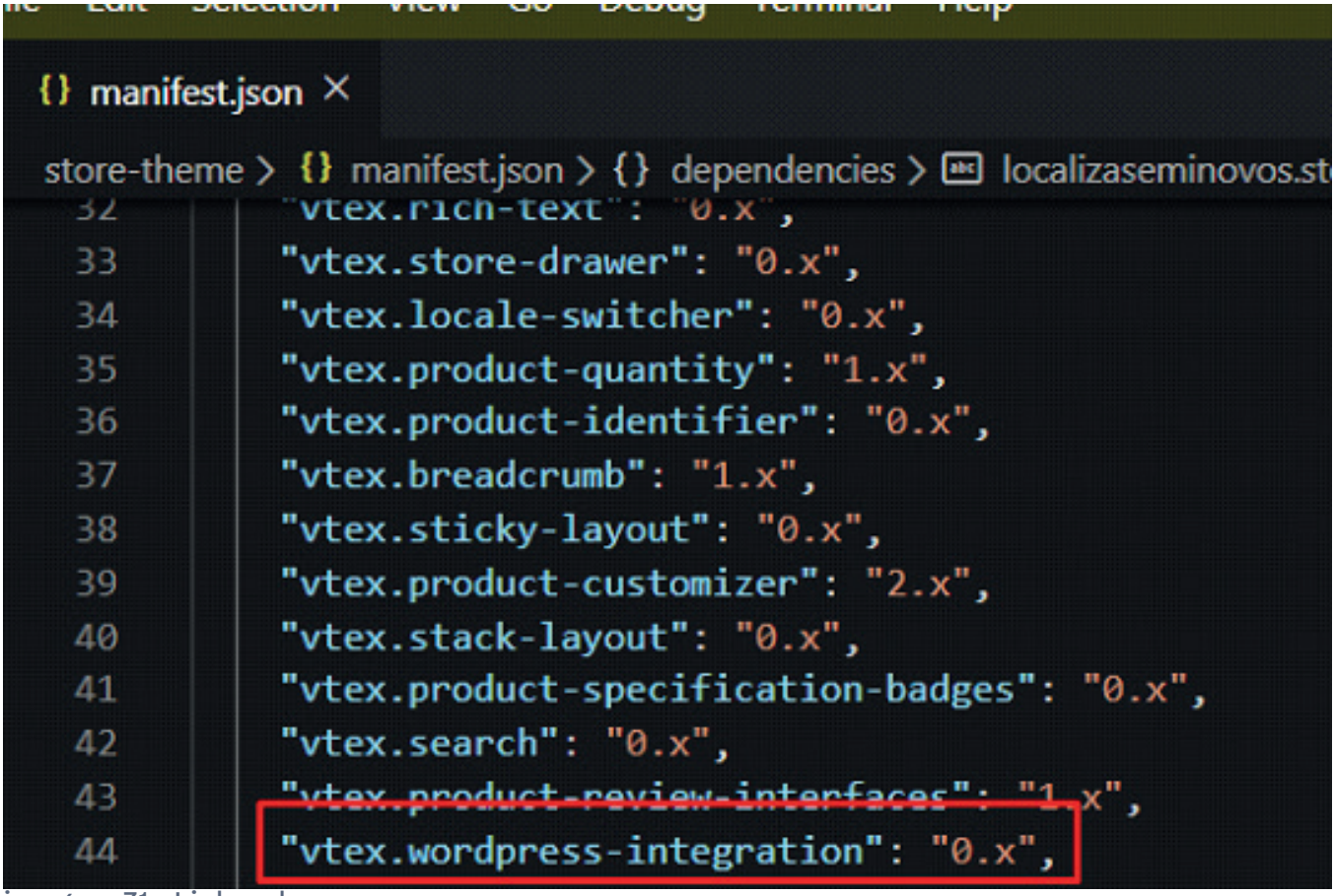


imagem 31 - Linkando um app

Após isso, basta seguir as opções de utilização do app dentro da documentação da VTEX.

5.4 Criando e validando um app

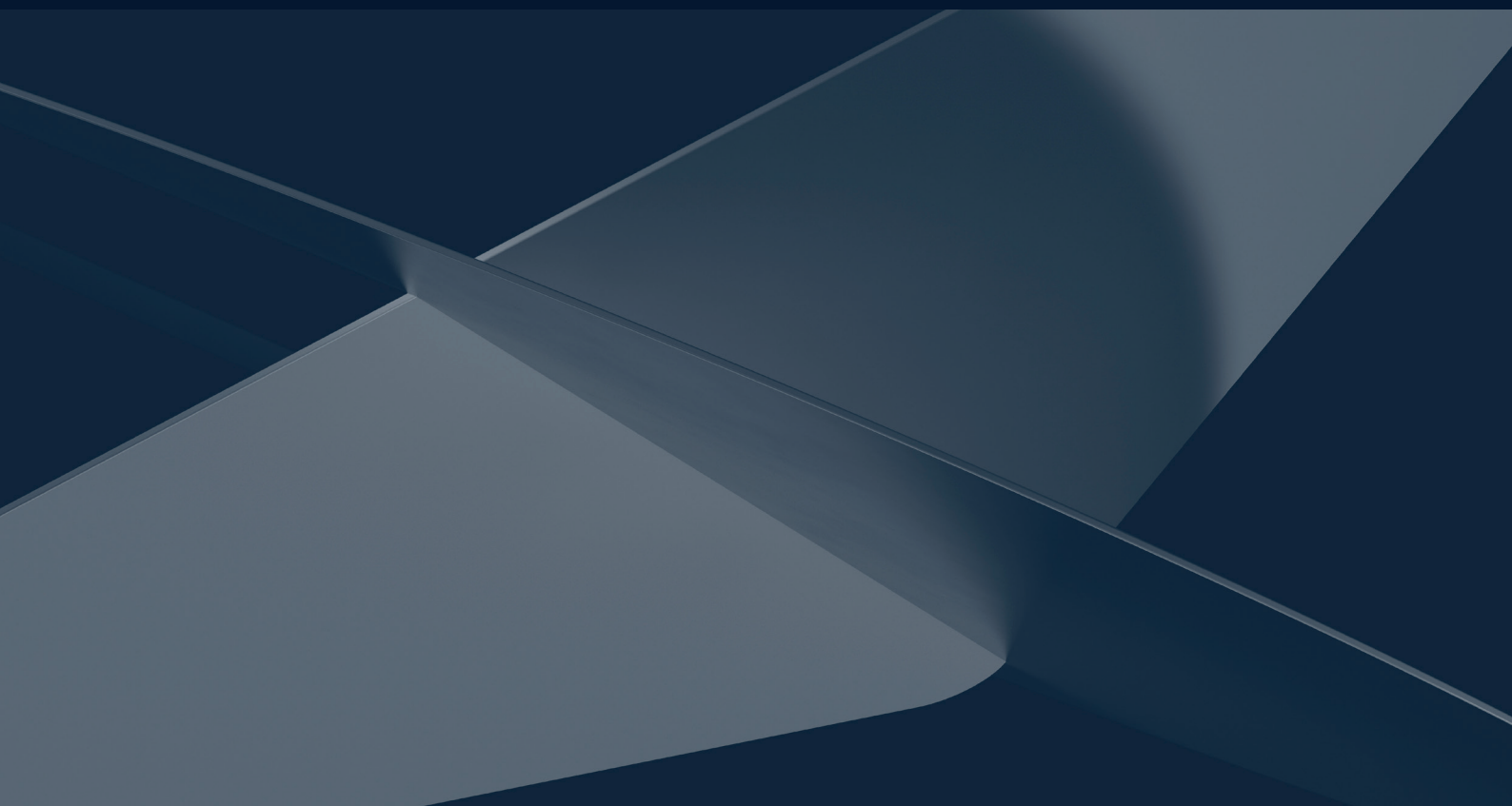
Linkar componentes e apps não irão garantir que sua loja funcione no ambiente final. Para isso, é preciso publicar seu app.

Basta usar o comando `vtex publish` para que o app seja disponibilizado e instalado na loja. Caso não precise de uma política de preços, será possível instalar de forma simples o app na loja.

Basta usar o comando `vtex install {vendor}.{appName}`, onde vendor é, geralmente, o account da loja e o appName é o nome do app que foi definido em seu manifest.

06

Graphql



6.1 O que é o GraphQl

O GraphQL é uma linguagem de consulta criada pelo Facebook.

Para fazer uma comparação e compreender melhor, podemos colocá-la ao lado do SQL, que é utilizado para consulta e manipulação de bancos de dados relacionais. O GraphQL, por sua vez, é utilizado para consulta e manipulação de APIs Web, por exemplo, mas não restrita somente a isso.

6.2 Trabalhando com GraphQl em sua loja

A maneira mais rápida e fácil de se trabalhar com o GraphQL em sua loja é instalando a extensão Apollo no Chrome.

imagem 32 - GraphQl

Ao instalar essa extensão, vá até a sua loja. Nesse exemplo, vamos acessar a página de produtos e abrir as ferramentas do desenvolvedor/console.

Repare que uma aba chamada “Apollo” surge como opção. Ao clicar nela, você encontrará ferramentas de consultas para integração onde houver necessidade:

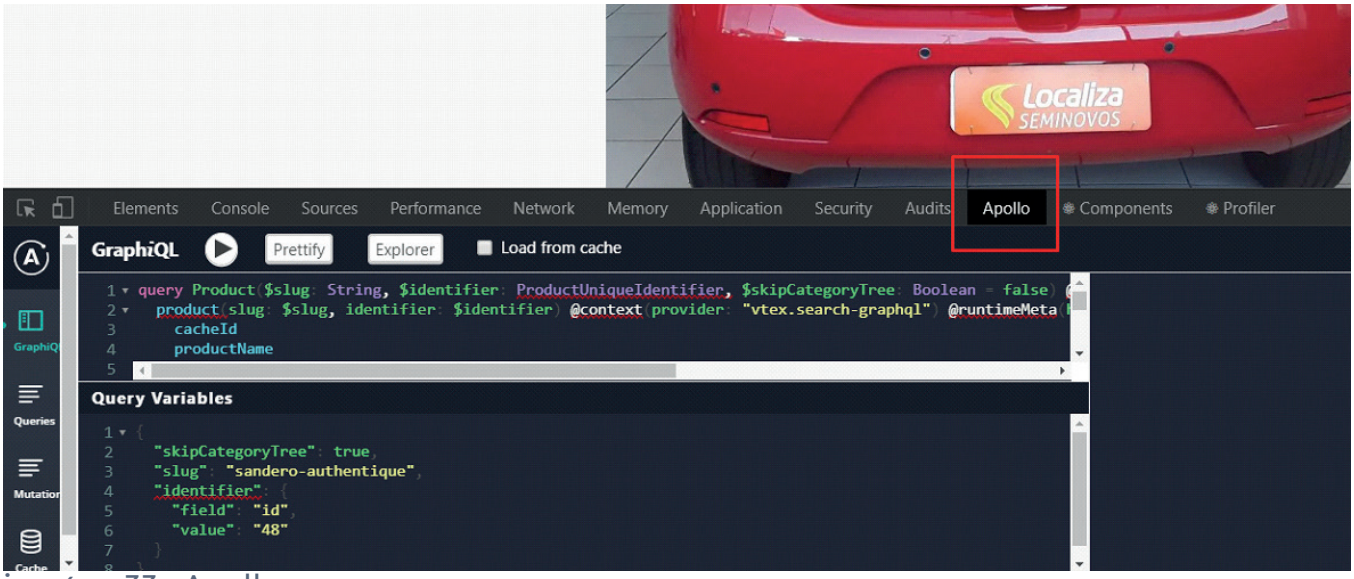


imagem 33 - Apollo

Para verificar quais consultas você pode utilizar naquela página, basta escolher a aba “Queries” e clicar no botão “Run in GraphQl”, conforme o print abaixo:

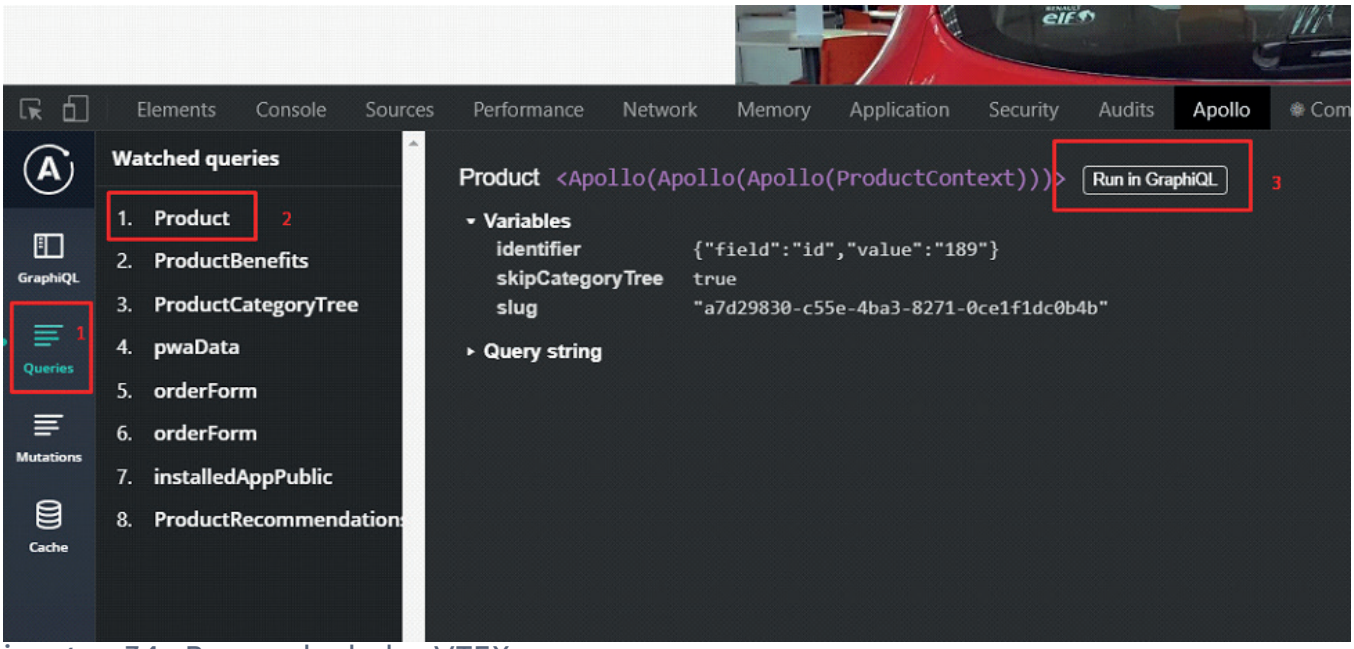


imagem 34 - Buscando dados VTEX

Desta forma, você será capaz de encontrar todos os dados que a VTEX retorna para utilizá-los da forma que preferir.

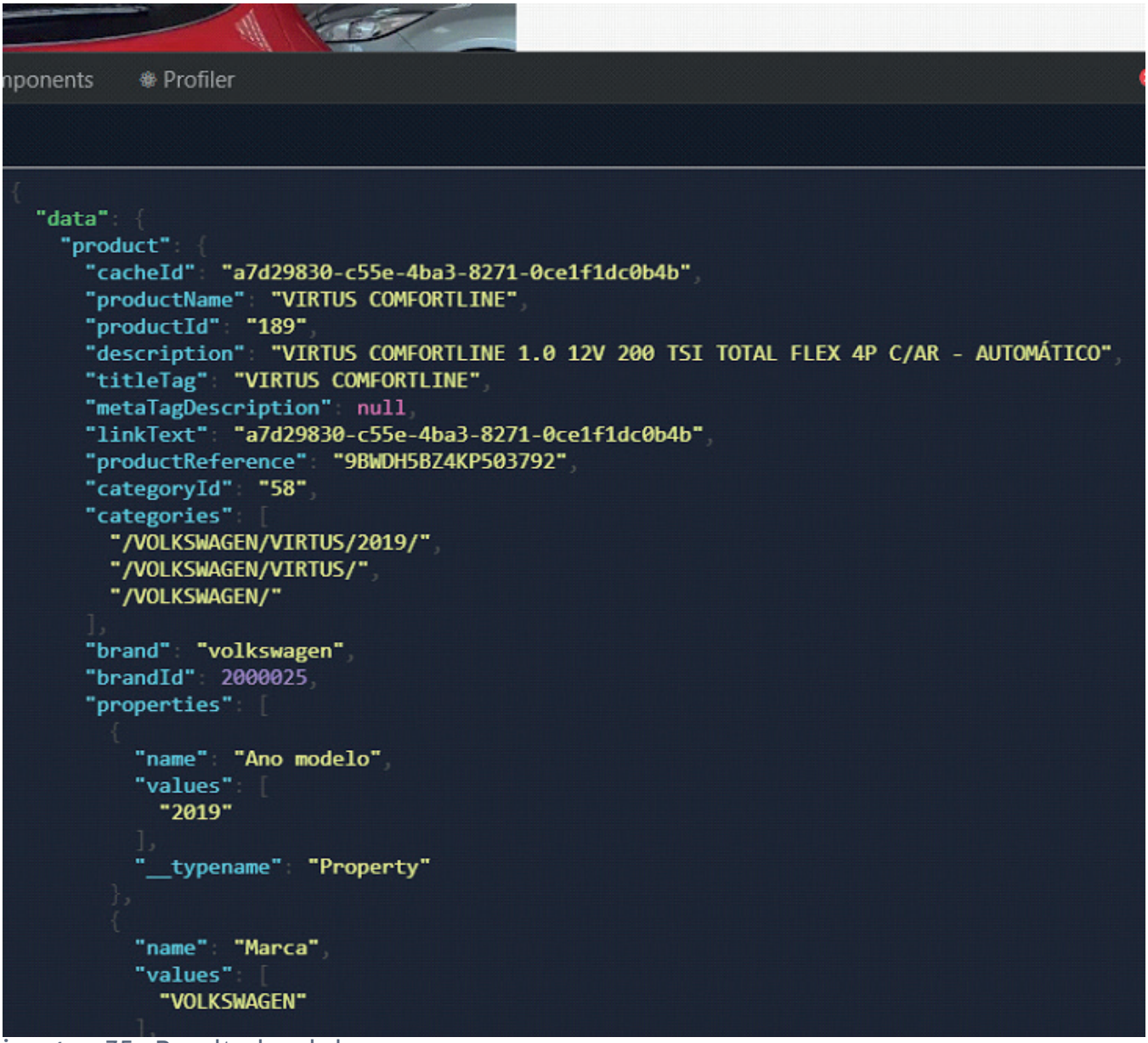


imagem 35 - Resultados da busca

6.3 Publicando sua loja

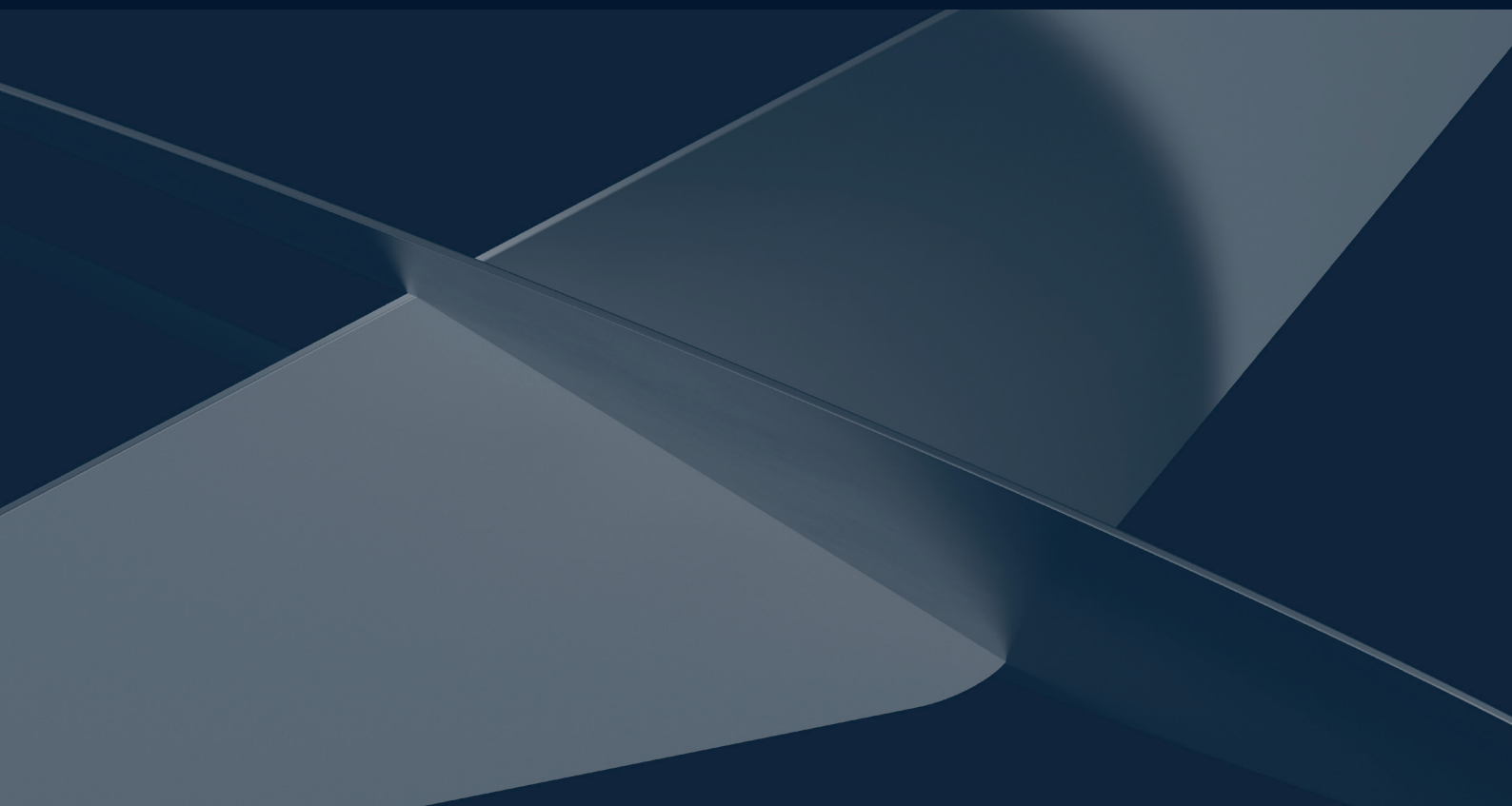
Agora que já programou, testou e corrigiu supostos erros da sua loja, é hora de publicá-la no workspace master!

É simples. Basta rodar o comando: vtex production e conferir se todos os apps, componentes e estilos estão funcionando corretamente nesse workspace.

Para finalmente colocar sua loja em produção, basta rodar o comando vtex workspace promote, tornando-o oficial, com o "workspace" sendo o nome do workspace que trabalhou desde o início e está pronto.

07

Administração



7.1 Site editor

Site Editor é a área onde não apenas o desenvolvedor, mas também o lojista pode editar dados, imagens, cores e outras possibilidades que possam ter sido desenvolvidas.

É de fácil utilização e compreensão.

7.2 Como utilizar o site editor?

Para acessar, basta ir em **Admin > CMS > Site Editor**

Ao clicar, repare que irá aparecer a loja ao lado, com uma coluna com todos os componentes criados dentro do blocks naquela página em específico.

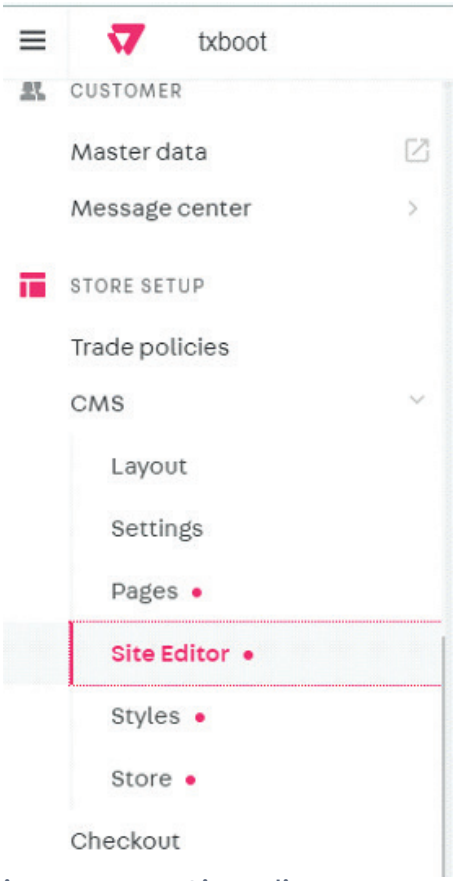


imagem 36 - Site Editor

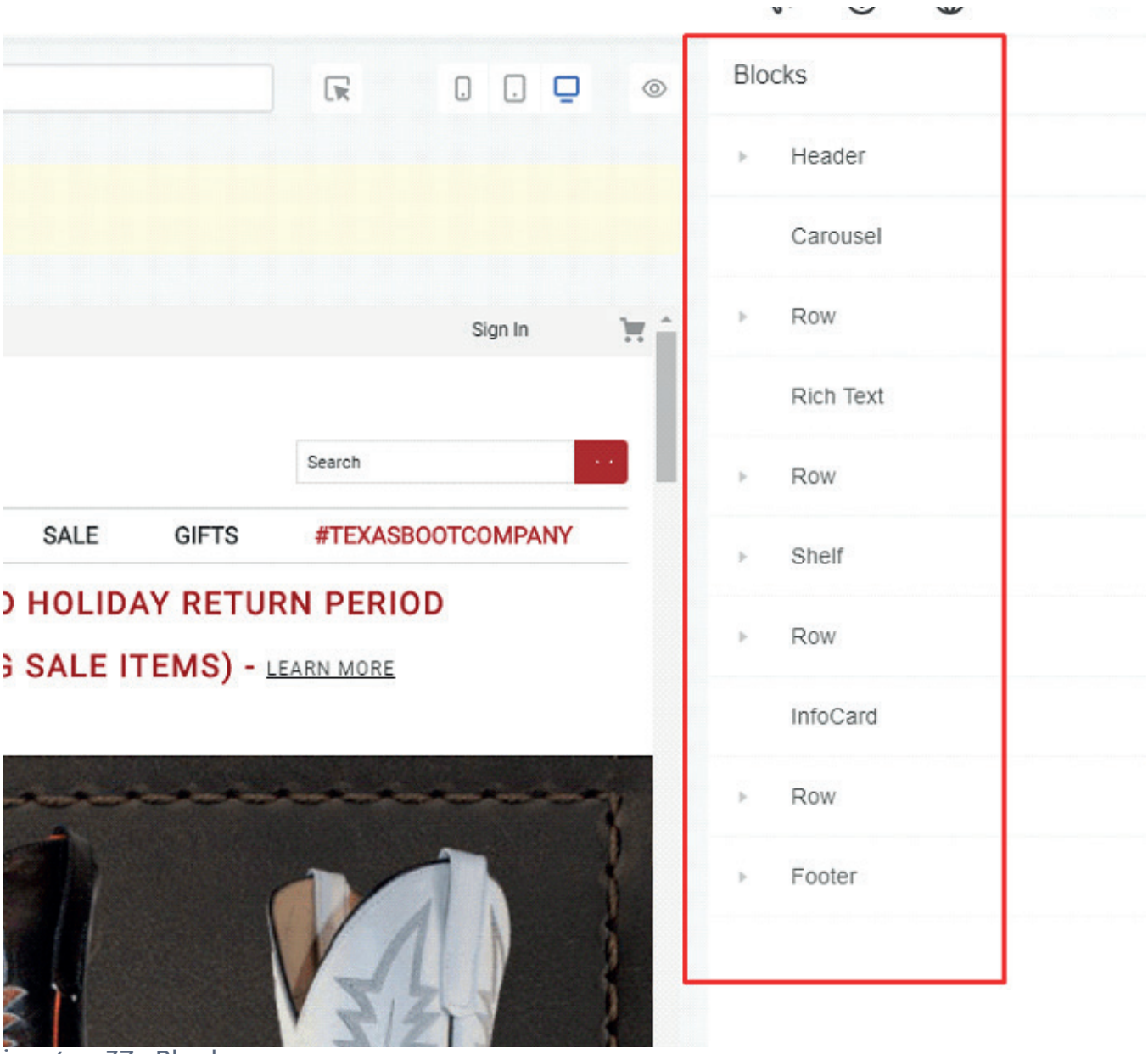


imagem 37 - Blocks

Cada uma das propriedades disponíveis para edição, que foram dadas como opções para o admin, aparecerão nessa área para o lojista.

7.3 Vinculando URL´s em banners

Como exemplo, o componente de carroussel é de fácil entendimento.

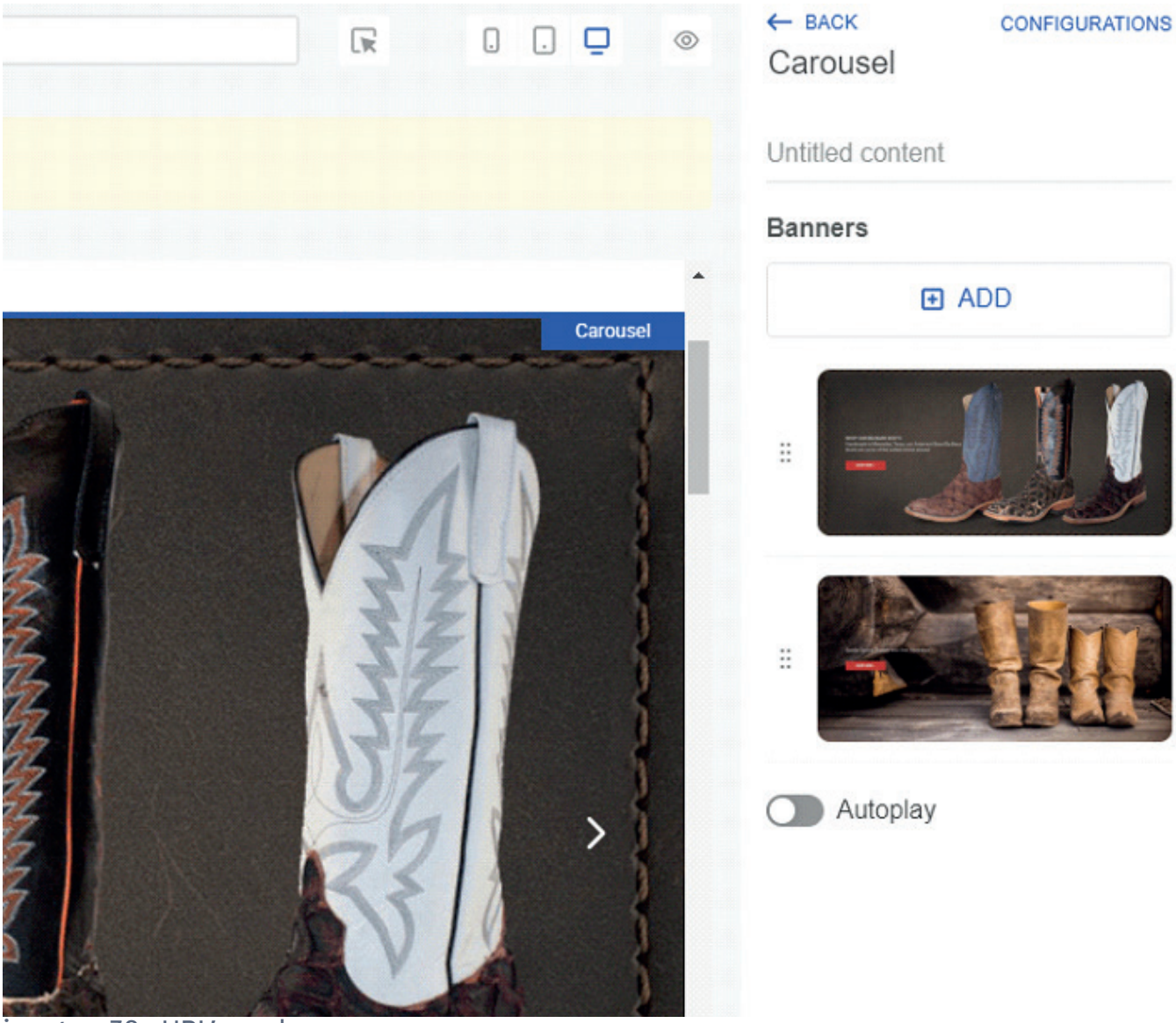


imagem 38 - URL's em banner

É possível inserir imagens para o banner da página. Você pode fazer as seguintes ações:

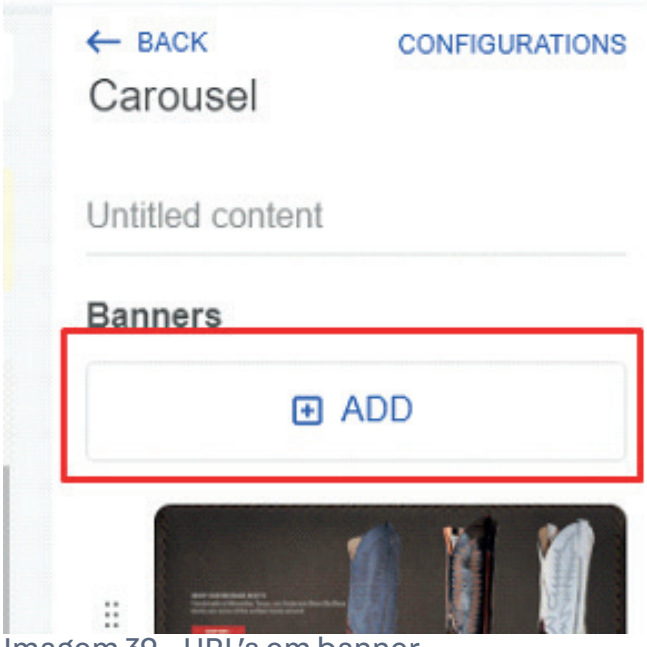


Imagem 39 - URL's em banner

Clicar no ícone de “seis pontos” para alterar a sequência dos banners.

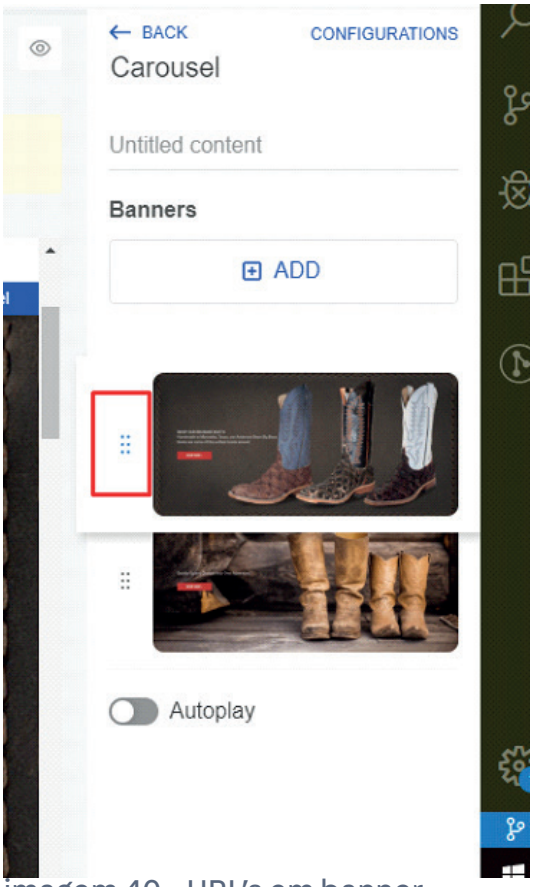


imagem 40 - URL’s em banner

E, ao passar o mouse sobre a imagem, o botão com opções de editar/deletar o banner.

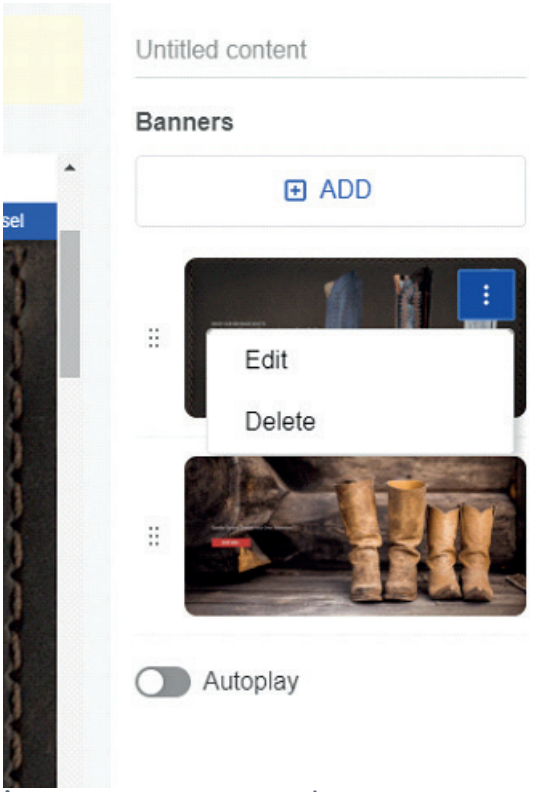


imagem 41 - URL’s em banner

Ao clicar em “Editar”, você pode mudar as configurações do banner, como nome, assim como inserir/alterar imagem para o mobile e o link, por exemplo.

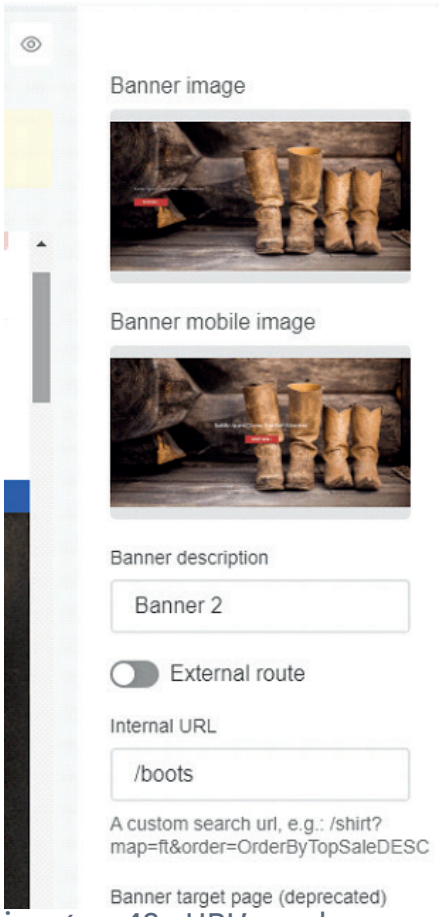


imagem 42 - URL’s em banner

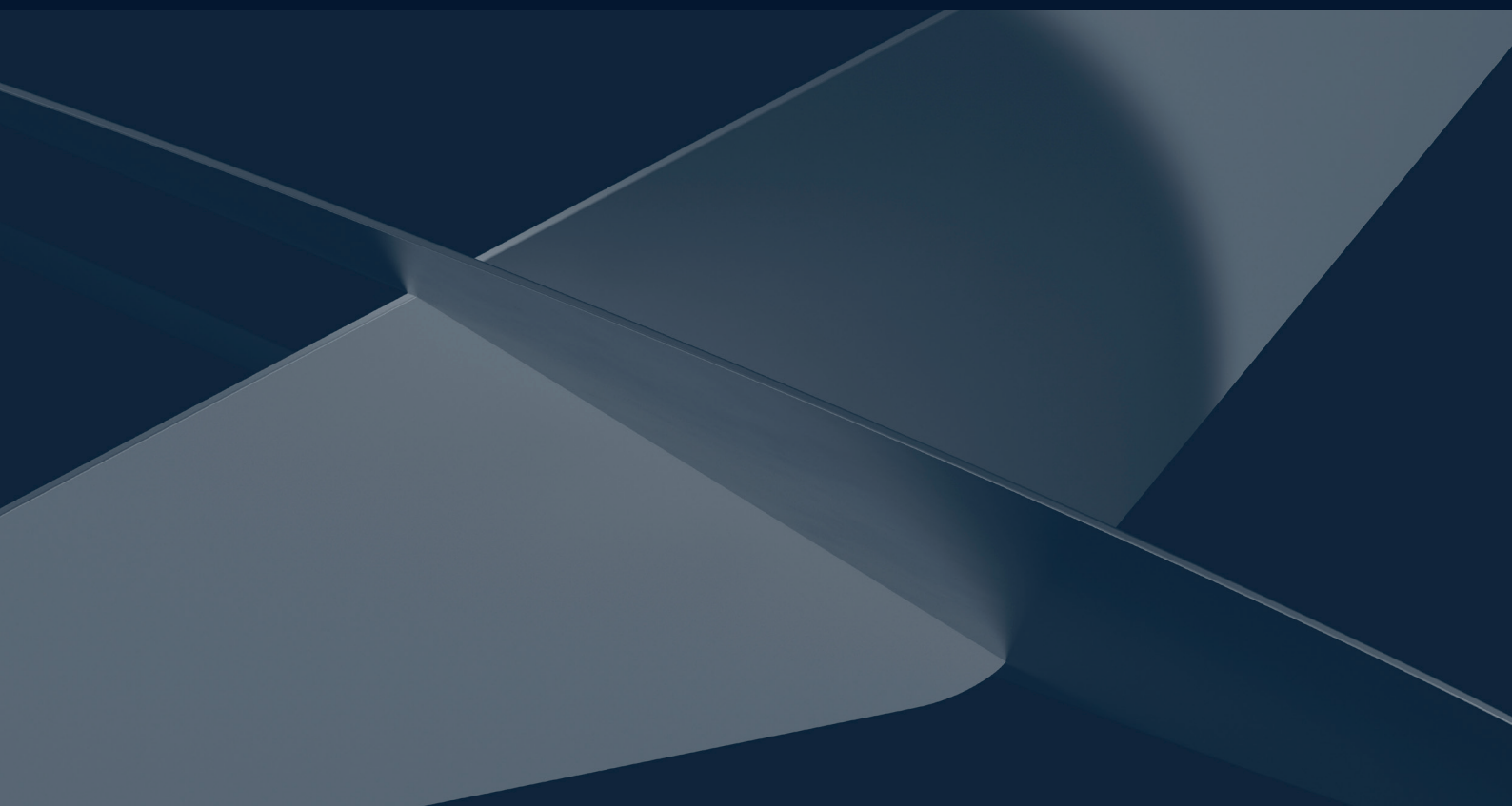
7.4 Recomendações

Conhecimento básico na linguagem React e ES6/JS é recomendado para melhor desenvolvimento na plataforma. Portanto, procure cursos que te ajudem a desenvolver tais habilidades.

A VTEX disponibiliza um curso no Github Learning para quem estiver procurando aprender o básico do VTEX IO de forma mais dinâmica, com correções automáticas feitas por meio de um boot. O curso está disponível neste link.

08

Case
TEXAS BOOT



Texas Boot

Desenvolvimento da loja Texas Boot Company para dentro do VTEX IO.

Sobre a marca

Fundada em 2010, a Texas Boot Company começou com a realização do sonho dos sócios Marc e Shelly Conselman de disponibilizar um estilo autêntico, com extrema qualidade e que se encaixasse confortavelmente ao traje típico ocidental, algo que tanto amavam.

8.1 Desafio

Em busca de atrair empresas americanas para dentro da plataforma, a VTEX optou por priorizar marcas significativas, como a Texas Boot Company.

Com o VTEX IO, fomos capazes de iniciar o desenvolvimento da loja online, tendo em base o layout e o conteúdo da loja atual da marca.

A necessidade era passar toda a identidade, cores, estruturas e conteúdos que já existiam na loja antiga com exatidão para dentro da plataforma.
Para tal, os desenvolvedores da Maeztra trabalharam em conjunto com a equipe VTEX. O case também possibilitou que fossem desenvolvidas melhorias para componentes do IO, como, por exemplo, no componente menu.

8.2 Resultados

See more at: vtex.com

Accelerate
Commerce
Transformation