

## CSS handles e o poder da customização de blocos

SUGGEST EDITS

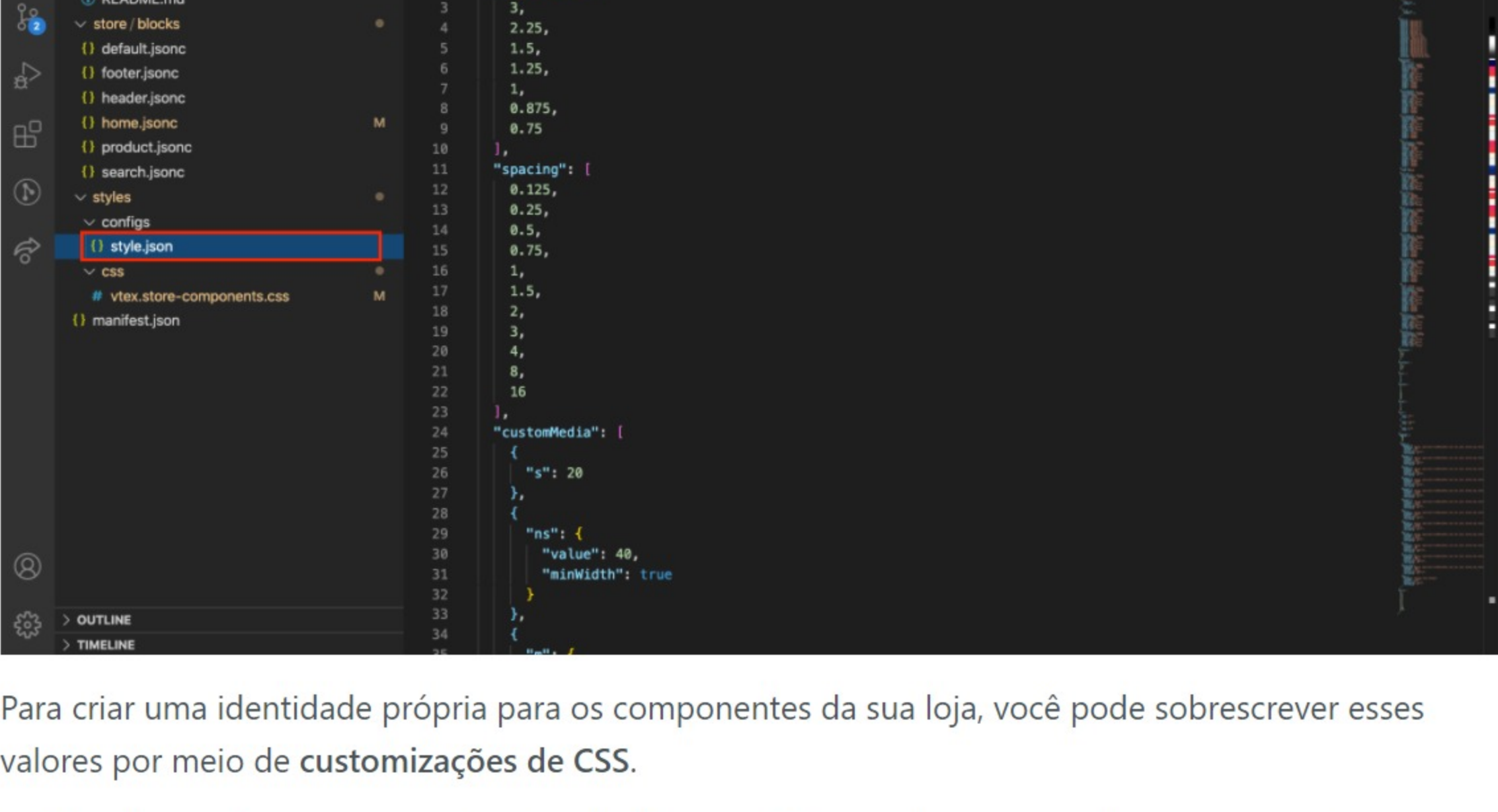
### CSS Handles e o poder da customização de blocos

#### Introdução

Dando uma rápida olhada na sua loja atual, você conseguirá perceber que todos os componentes possuem estilos parecidos, mesmo que nenhuma customização tenha sido feita por você.

Todos eles compartilham valores **pré-estabelecidos** para fonte, cor de fundo, cor principal, formato dos botões, etc. Por exemplo, o [Info Card](#).

Isso se deve ao `style.json`, arquivo responsável por declarar valores genéricos de customização para toda loja do Store Framework.



Para criar uma identidade própria para os componentes da sua loja, você pode sobrescrever esses valores por meio de **customizações de CSS**.

Analisando a [receita](#) para customizações de loja por CSS, percebemos que alguns passos serão necessários para aplicar o estilo próprio desejado por você, como:

1. Criar um novo arquivo dentro da pasta `css` com o nome `vtex-{{AppName}}.css`.
2. Usar o **CSS Handle** do componente que será customizado dentro deste arquivo seguindo o formato abaixo:

```
.{{CSSHandle}} {
  {{PropriedadeDeCSS}}: {{ValorDesejado}};
  {{PropriedadeDeCSS}}: {{ValorDesejado}};
}
```

3. Na falta de **CSS Handles**, aplicar **CSS Selectors** permitidos, como é o caso do `:global(vtex-{{componentName}})`.
4. Para aplicar CSS em um bloco específico e não a todos os blocos daquele tipo, usa-se o recurso de `blockClass`, que aparece ao lado dos **handles** de css ao inspecionar seu código. As `blockClass` devem ser declaradas como uma prop no bloco em questão, e então referenciado no arquivo de estilo como mostrado abaixo:

```
.{{CSSHandle}}--{{blockClass}} {
  {{PropriedadeDeCSS}}: {{ValorDesejado}};
  {{PropriedadeDeCSS}}: {{ValorDesejado}};
}
```

#### Adicionando Info Cards ao *Minimum Boilerplate*

Antes de customizar um *info card* é necessário adicioná-lo ao tema, dado que o *boilerplate* é muito simples. Com isso em mente, vá ao arquivo `home.jsonc` e adicione dois *info cards* aos seus blocos:

```
{
  "store.home": {
    "blocks": [
      "rich-text",
      "info-card#clearance",
      "info-card#vintage"
    ],
    "rich-text": {
      "props": {
        "text": "VTEX Store Framework",
        "textAlignment": "CENTER",
        "textPosition": "CENTER",
        "font": "t-heading-1"
      }
    }
  }
}
```

Agora, no mesmo arquivo, vamos definir os blocos logo abaixo da definição de `rich-text`. Você pode utilizar o código abaixo para inserir no arquivo:

```
"info-card#clearance": {
  "props": {
    "id": "info-card-clearance",
    "isFullModeStyle": false,
    "textPosition": "left",
    "imageUrl": "https://storecomponents.vteximg.com.br/arquivos/banner-info-card-clearance.jpg",
    "headline": "Clearance Sale",
    "callToActionText": "DISCOVER",
    "callToActionUrl": "/sale/d",
    "textAlignment": "center"
  }
},
"info-card#vintage": {
  "props": {
    "id": "info-card-vintage",
    "isFullModeStyle": false,
    "textPosition": "right",
    "imageUrl": "https://appliance.vteximg.com.br/arquivos/cozinha-rosa.jpg",
    "headline": "Vintage",
    "callToActionText": "DISCOVER",
    "callToActionUrl": "/sale/d"
  }
}
```

#### Customizando o Info Card

Para descobrir os **CSS Handles** de um componente, como o *Info Card*, basta acessar a [seção Customization](#) da sua documentação.

De acordo com a descrição dos **CSS Handles** e com a receita customizações de loja por CSS, conseguimos implementar um exemplo de *Info Card* customizado.

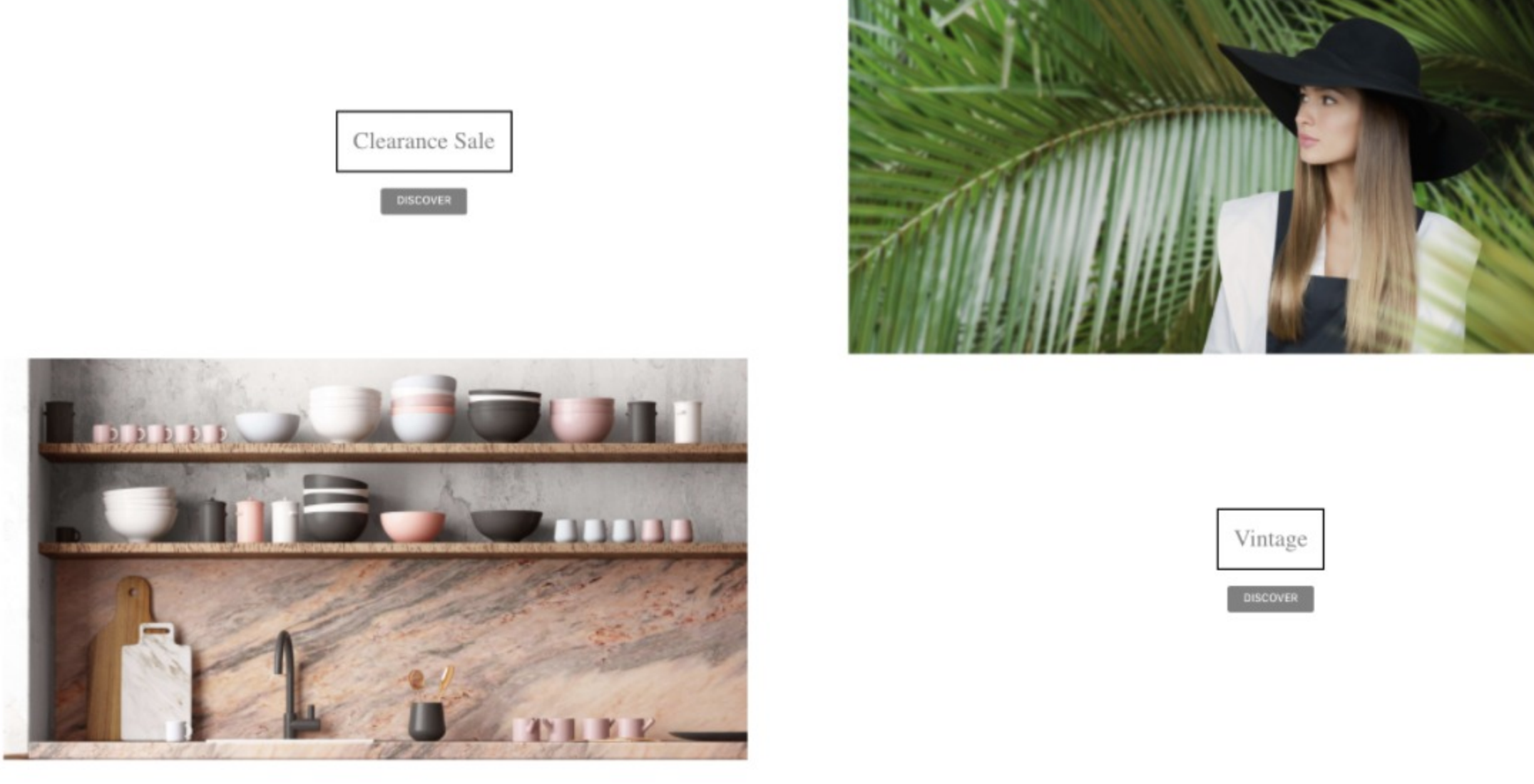
1. No arquivo `vtex.store-components.css` dentro de `/styles/css`, vamos alterar seu título e as configurações do botão *call to action*, adicionando o código a seguir:

```
/* /styles/css/vtex.store-components.css */
.infoCardHeadline {
  font-family: serif;
  font-size: 2.25rem;
  font-weight: normal;
  color: gray;
  border: 2px solid black;
  padding: 24px;
}

.infoCardCallActionContainer :global(.vtex-button) {
  color: white;
  background-color: gray;
  border: transparent;
}

.infoCardCallActionContainer :global(.vtex-button):hover {
  color: gray;
  background-color: blue;
  border: transparent;
}
```

Você pode conferir o efeito das mudanças feitas por você executando o comando `vtex link`.

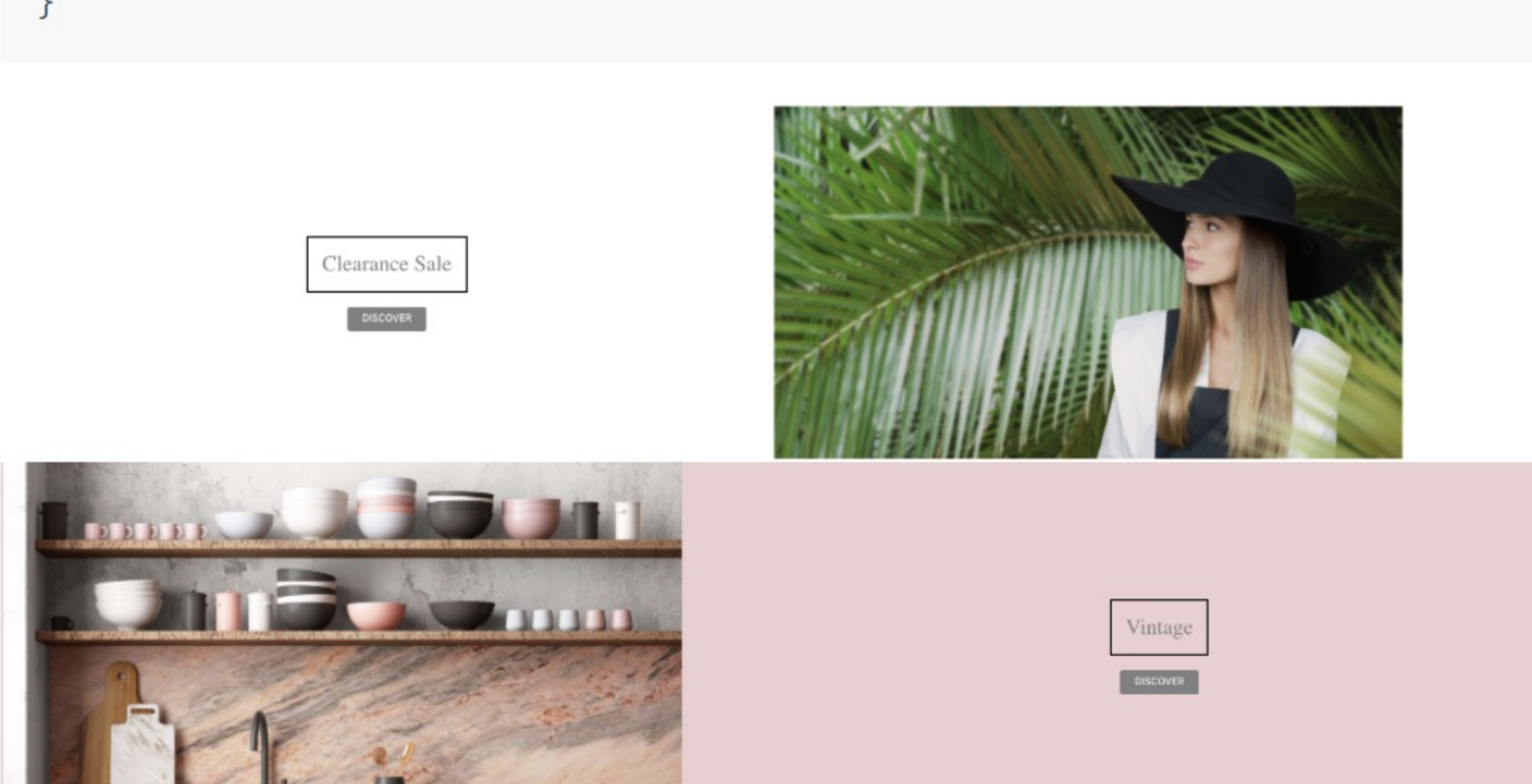


2. Em seguida, vamos adicionar um estilo específico para o *Info Card Vintage*. Para isso, precisamos adicionar a propriedade `blockClass` ao info card, como mostrado a seguir:

```
"info-card#vintage": {
  "props": {
    "id": "info-card-vintage",
    "isFullModeStyle": false,
    "textPosition": "right",
    "imageUrl": "https://appliance.vteximg.com.br/arquivos/cozinha-rosa.jpg",
    "headline": "Vintage",
    "callToActionText": "DISCOVER",
    "callToActionUrl": "/sale/d",
    "textAlignment": "center",
    "blockClass": "vintage"
  }
}
```

3. E então declare uma `background-color` para este *Info Card* específico no seu arquivo de CSS:

```
/* /styles/css/vtex.store-components.css */
.infoCardContainer--vintage {
  background-color: #edcfd1;
}
```

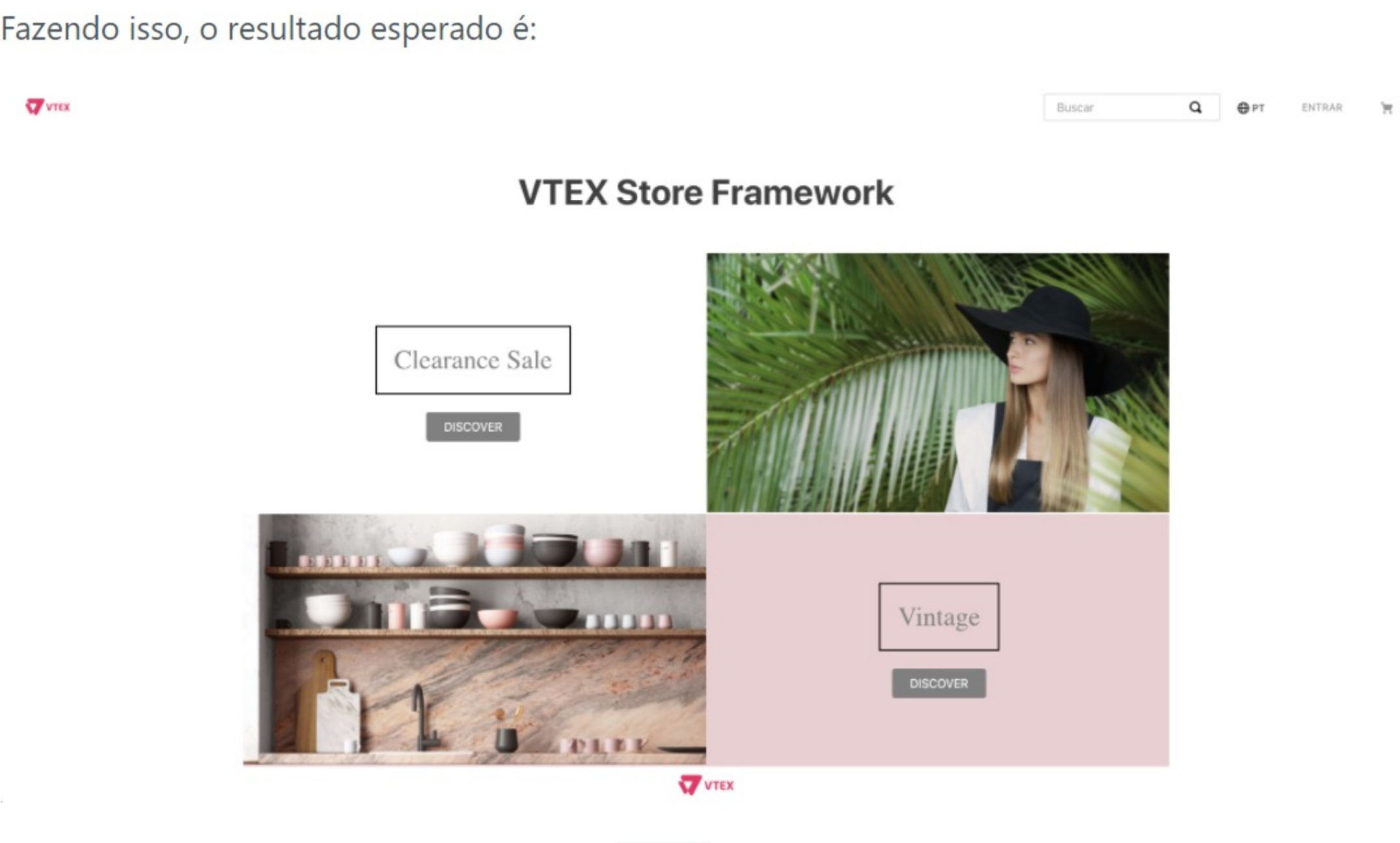


4. Agora, vamos finalizar adicionando mais estilos customizados para o nosso *Info Card*. No mesmo arquivo CSS que foi utilizado anteriormente, defina a largura máxima (`max-width`) de todos os infocards para `1260px`, a margem (`margin`) para `0 auto` e o espaçamento (`padding`) para `0`.

Você pode se basear nos **Handles** do [Info Card](#).

```
/* /styles/css/vtex.store-components.css */
.infoCardContainer {
  max-width: 1260px;
  margin: 0 auto;
  padding: 0;
}
```

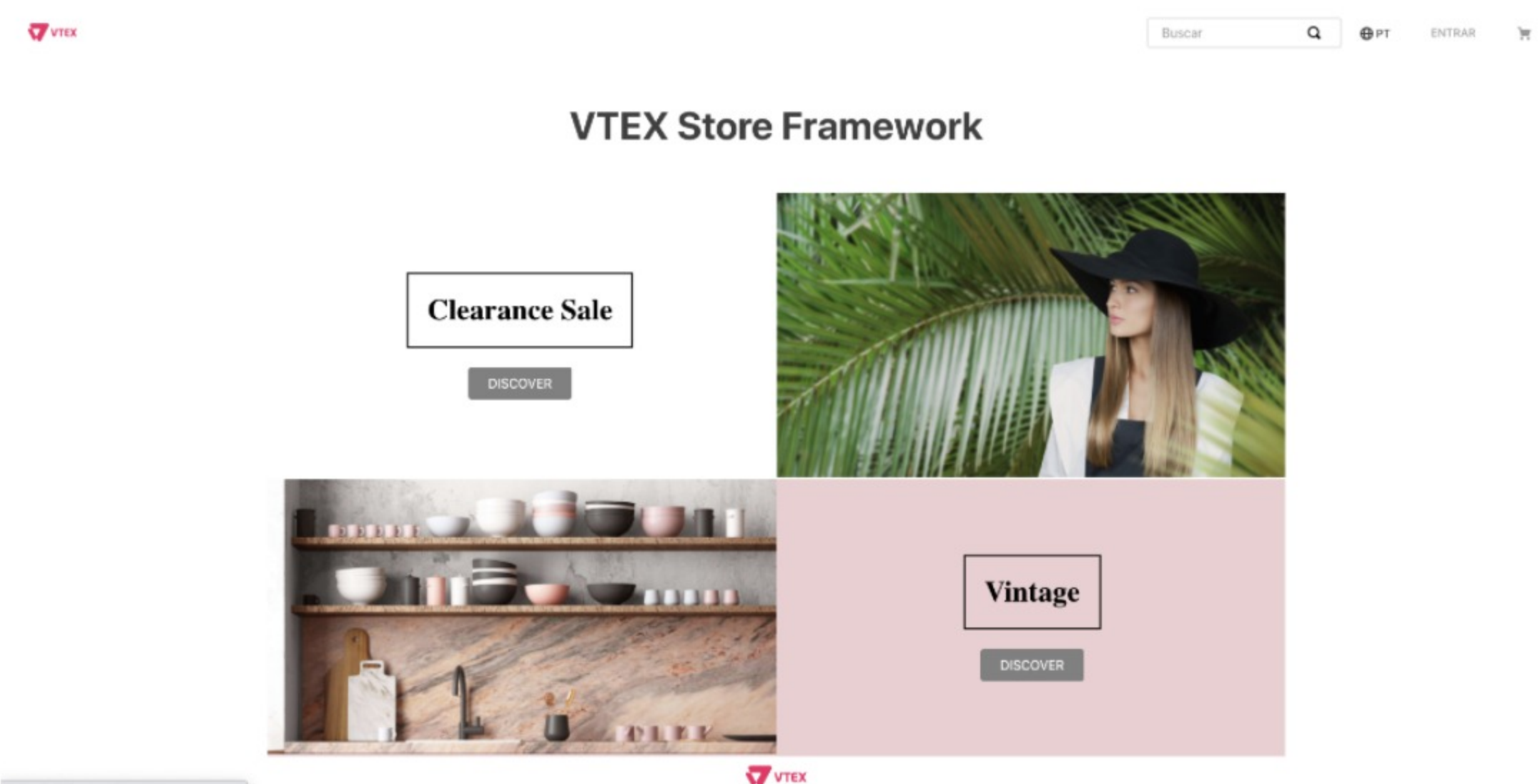
Fazendo isso, o resultado esperado é:



5. Mude a cor do título do componente para `black` e em negrito:

```
.infoCardHeadline {
  font-family: serif;
  font-size: 2.25rem;
  font-weight: bold;
  color: black;
  border: 2px solid black;
  padding: 24px;
}
```

Terminado este passo, você teve ter algo similar a:



6. Mude a cor de fundo do botão durante o *hover* para `white`:

```
.infoCardCallActionContainer :global(.vtex-button):hover {
  color: gray;
  background-color: white;
  border: transparent;
}
```

Certifique-se de testar também a ação de *hover*, de forma a validar se as cores para esta ação também estão corretas.

Está com dúvidas?

Veja o [gabarito para esta etapa](#) ou participe do nosso [office hours](#)

Ajude-nos a fazer este conteúdo melhor!

Os cursos do VTEX IO são de código aberto. Se você perceber algum problema, pode abrir um pull request!

[FAÇA UMA CONTRIBUIÇÃO](#)

ou [abra uma issue](#)

Updated 23 days ago

Próximo passo

Estilos globais