

Teste Prático - Frontend

Bem-vindo à prova prática para candidatos à área de front-end do Mercado Livre.

Apresentamos o design e a descrição funcional de um pequeno aplicativo que será a base do trabalho que você deve desenvolver.

O aplicativo consiste em três componentes principais: a caixa de pesquisa, a exibição de resultados e a descrição dos detalhes do produto.

Você deve usar o seguinte stack tecnológico para construir o aplicativo:

- Cliente:
 - HTML
 - JS (desejável usar React ou Backbone)
 - CSS (desejável usar Sass)
- Servidor:
 - Node >= 6
 - Express

Para resolvê-lo, pedimos que você leve em consideração:

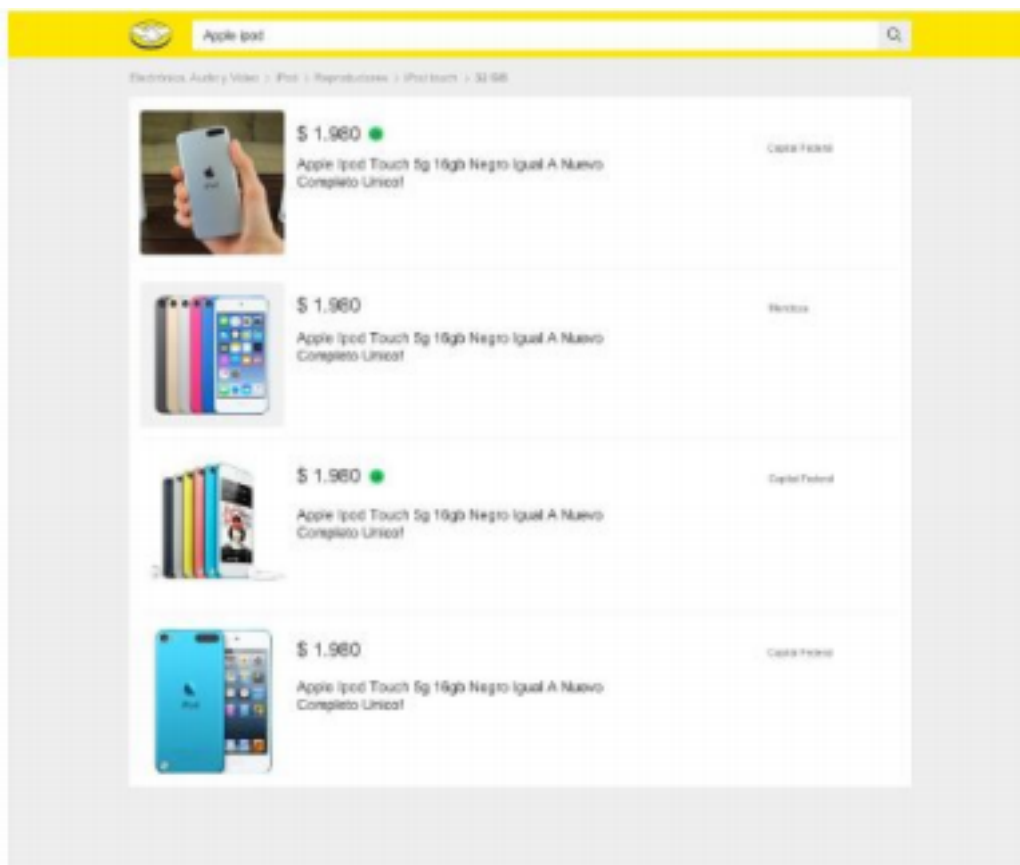
- Usabilidade
- SEO
- Performance
- Escalabilidade

Primeiro, apresentamos as visualizações das telas que você precisa fazer:

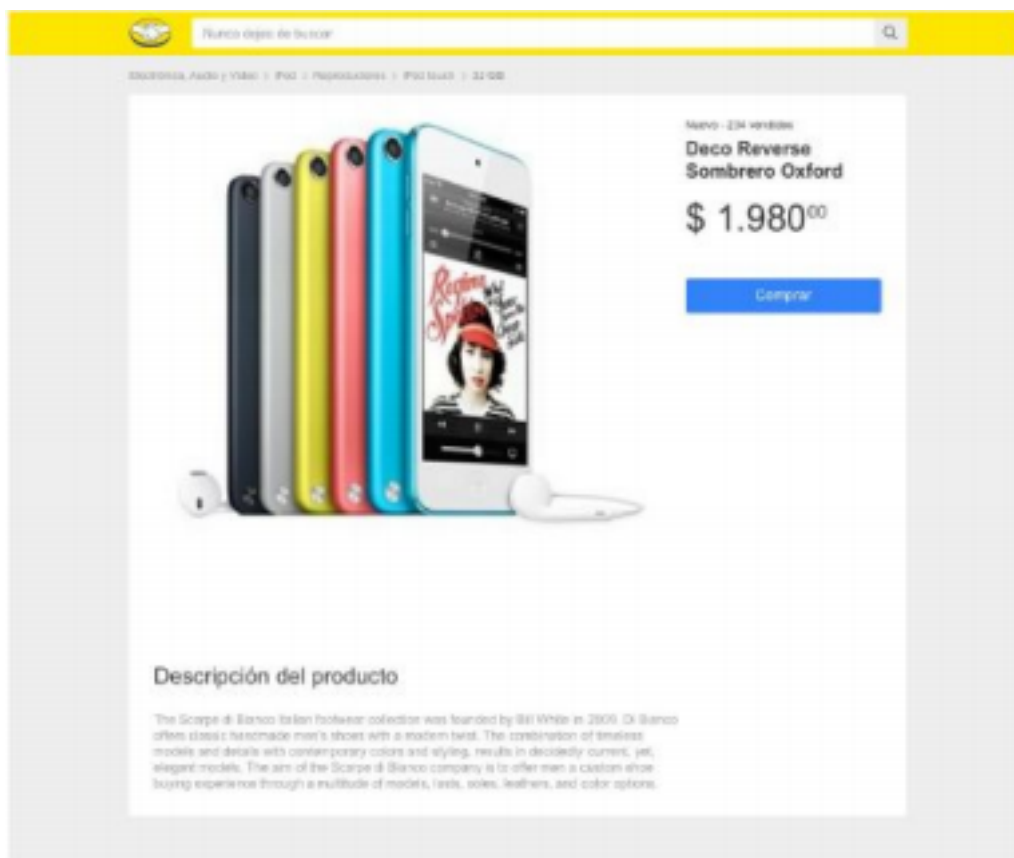
1. Caixa de pesquisa



2. Resultado da pesquisa



3. Detalhes do produto



Pedimos a você:

- Com base nos designs fornecidos, construa as três visualizações a seguir:
 - Caixa de pesquisa
 - Resultados da pesquisa
 - Detalhe do produto

- As visualizações são navegáveis de forma independente e têm seu próprio url:
 - Caixa de pesquisa: "/"
 - Resultados da pesquisa: "/ items? Search ="
 - Detalhes do produto: "/ items /: id"
- Crie os seguintes endpoints para serem usados nas visualizações:
 - / api / items? Q =: **query**

■ Você deve ver o seguinte endpoint:

<https://api.mercadolivre.com/sites/MLA/search?q=: query>

E retornar os resultados no formato indicado:

```
{
  "autor": {
    "name": String
    "lastname": String
  },
  categories: [String, String, String, ...],
  items: [
    {
      "id": String,
      "title": String,
      "price": {
        "currency": String,
        "amount": Number,
        "decimals": Number
      },
      "picture": String,
      "condition": String,
      "free_shipping": Boolean
    },
    {...},
    {...},
    {...}
  ]
}
```

○ / api / items /: **id**

■ **Você** deve ver os seguintes endpoints:

<https://api.mercadolivre.com/items/: id>
<https://api.mercadolivre.com/items/: id/ descrição>

E retornar os resultados no formato indicado:

```
{
  "author": {
    "name": String
    "lastname": String
  },
  "item": {
    "id": String,
```

```
"title": String,
"price": {
  "currency": String,
  "amount": Number,
  "decimals": Number,
},
"picture": String,
"condition": String,
"free_shipping": Boolean,
"sold_quantity": Number
"descrição": String
}
}
```

Descrição funcional do aplicativo

- Na visualização da caixa de pesquisa, você deve conseguir inserir o produto a ser pesquisado e, ao enviar o formulário, vá para a visualização "Resultados da pesquisa" exibindo apenas 4 produtos. Então, clicando em um deles, você deve navegar para a visualização de detalhes do produto.
- Com a identificação de um produto, você deve ser capaz de ir diretamente para a exibição de detalhes do produto.

Entregável

- Repositório no Github.
- Envie o link para o repositório por e-mail.

Notas

- A assinatura json no campo do autor refere-se ao seu nome e sobrenome. Você deve adicionar esta assinatura no tratamento dados entre a API e o front-end.
- O repositório pode ser público ou privado, conforme sua preferência.
- Não é necessário implementar paginação ou filtrar por categorias.
- O breadcrumb mostrado na lista de pesquisa deve ser montado com base na categoria que mais resultados obtidos (esta informação está disponível na API de pesquisa). (Obviamente, o breadcrumb da página de detalhes do item deve ser armada com a própria categoria do item).
- Você pode usar na lista de pesquisa a imagem que a API retorna (90 x 90), mesmo que pareça pixelada quando estique-o para se ajustar ao design do teste. (No final do teste, você não precisa procurar a imagem no tamanho exato).