

Teste Prático - Frontend

Bem vindo ao teste prático para candidatos frontend para o Mercado Livre.

A seguir, apresentamos o design e a descrição funcional de uma pequena aplicação que será a base do trabalho que deverá desenvolver.

A aplicação consiste em três componentes principais: A caixa de pesquisa, a visualização dos resultados e a descrição detalhada do produto.

Você deve usar a seguinte stack tecnológica para construir a aplicação:

- **Client:**
 - HTML
 - JavaScript (*Desejável utilizar React ou Backbone*)
 - CSS (*Desejável utilizar Sass*)
- **Server:**
 - Node ≥ 6
 - Express

Para resolver, pedimos que leve em consideração:

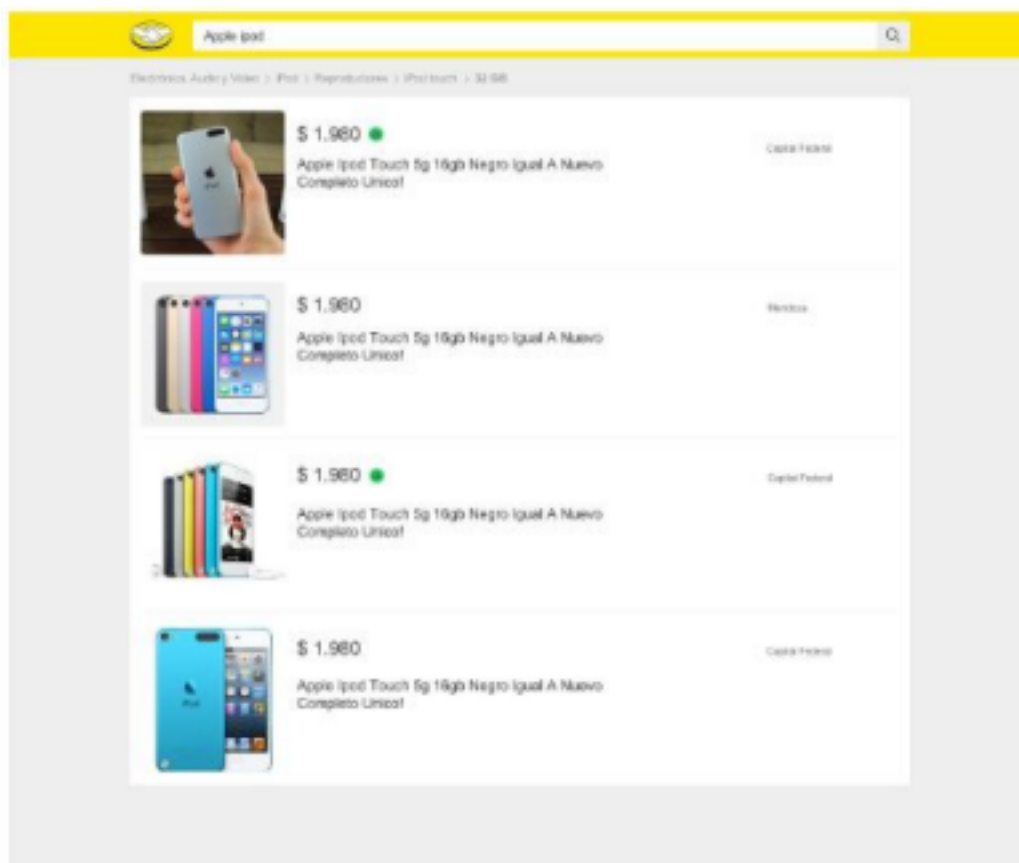
- Usabilidade
- SEO
- Performance
- Escalabilidade

Primeiro, te apresentamos as visualização das telas que você deve fazer:

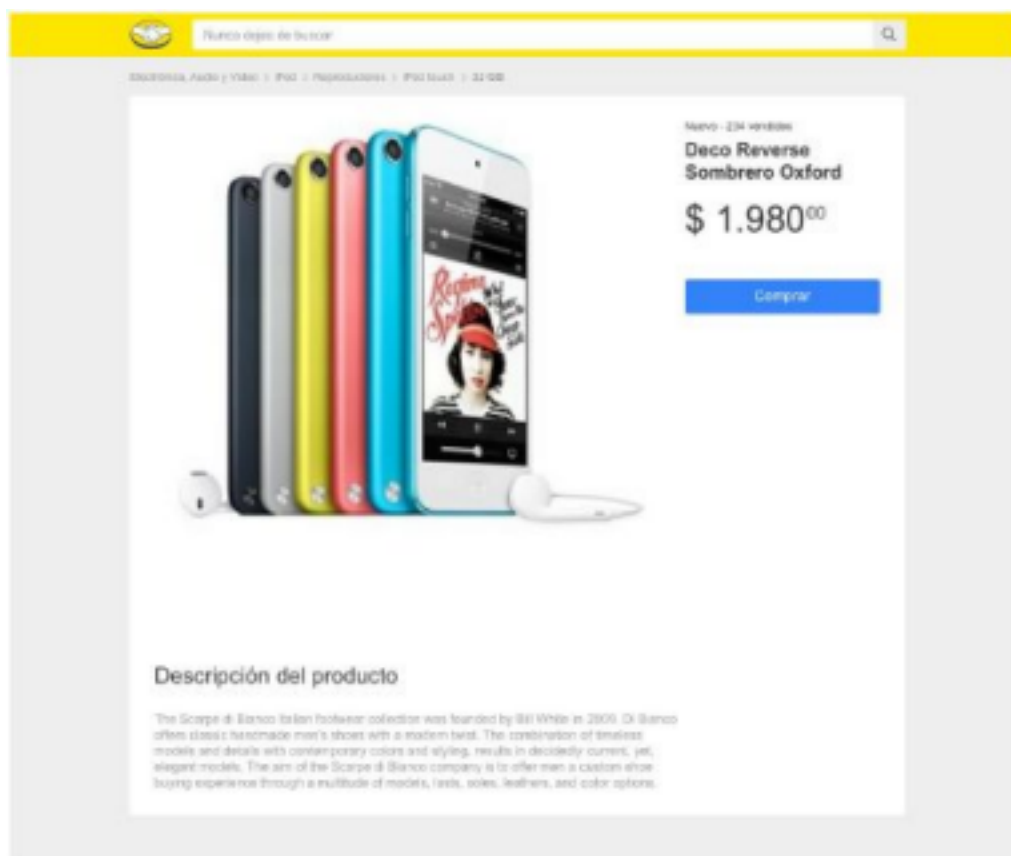
1. Caixa de pesquisa:



2. Resultado da pesquisa:



3. Detalhes do produto:



Te pedimos:

- Com base nos desenhos fornecidos, construir as seguintes visualizações:
 - Caixa de pesquisa

- Resultados da pesquisa
 - Detalhes do produto
- As visualizações são navegáveis de forma independente e possuem URL própria:
 - Caixa de pesquisa: `/`
 - Resultados da pesquisa: `/items?search=:search`
 - Detalhes dos produtos: `/items/:id`
 - Construir os seguintes endpoints para serem utilizados nas páginas:
 - `/api/items?q=:query`
 - Você deve utilizar o seguinte endpoint:
 - `https://api.mercadolivre.com/sites/MLA/search?q=:query`
 - E desenvolver os resultados no formato indicado:

JavaScript

```
{
  "author": {
    "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 consultar os seguintes endpoints:
 - `https://api.mercadolivre.com/items/:id`
 - `https://api.mercadolivre.com/items/:id/description`
 - E desenvolver os resultados no formato indicado:

JavaScript

```
{
  "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
    "description": String
  }
}
```

Descrição funcional da aplicação:

- Na visualização da caixa de pesquisa, você deverá conseguir inserir o produto a ser pesquisado e, ao enviar o formulário, navegar até a visualização de **resultados da pesquisa**, visualizando **somente 4 produtos**. Ao **clicar** em um deles, deverá levar até a visualização de **detalhes do produto**.
- Caso um usuário entre na URL que já contenha o ID do produto, deverá entrar diretamente na visualização de detalhes do produto.

Entrega do Teste:

- Repositório no Github.
- Enviar o link do repositório por e-mail.

Notas:

- O campo `author` no JSON, refere-se ao seu nome e sobrenome. Você precisará adicionar esta assinatura no tratamento de dados entre a API e o front-end.
- O repositório pode ser público ou privado, como você preferir.
- Não é necessário implementar paginação e filtro por categorias.
- O breadcrumb que aparece na lista de pesquisa deve ser montado com base na categoria que obteve mais resultados (essa informação está disponível na API de Search).
- O breadcrumb da página de detalhes do item deve ser criado com a própria categoria do item.
- Você pode usar a imagem retornada pela API na lista de pesquisa (90x90) embora pareça pixelado, quando esticada para se ajustar ao design do teste. (Para fins de teste, você pode encontrar a imagem no tamanho exato).