

PROJETO A3

Relatório

Equipe NESK:

Enrico Falcão - 1272216661

Ignácio Nunes Piva - 1272212685

Lucas Yann Santos Silva -1272215507

Lucas Ribeiro Santiago Nunes - 12722118292

Lúcio Santiago Marques de Queiroz – 12722123508

Relatório da Atividade da A3 (criação uma aplicação WEB para gerenciar catálogo de jogos de um usuário).

Salvador
2023

OBJETIVO

Este relatório tem como objetivo pontuar as principais informações do projeto, evidenciando nossa linha de raciocínio para o desenvolvimento do CRUD sinalizando os cumprimentos dos requisitos propostos pelos orientadores.

RESUMO

A aplicação web estará situada no site **Nesk Games**, tendo sua identidade visual pautada nas cores azul e roxo, e referenciada por um coelho, onde será a principal imagem da empresa. O site é composto por 6 páginas ao todo (Cadastro do Usuário, Login do Usuário, Cadastro do Jogo, Cadastro da Plataforma, Página Inicial e Edição de Jogo).

No processo inicial, começamos criando nossos primeiros traços por Wireframes, feitos pelo Paint, para alinharmos nossas ideias e gostos. Depois evoluímos para o Figma, criando corpo para o site e definindo a identidade e traduzimos para a realidade com o Javascript, HTML e CSS.

WIREFRAMES

Os modelos de Wireframes abaixo representam as a idealizações da plataforma, sendo:

1) Wireframe da Tela de Login:

Obs: o espaço azul deverá ser a logo.

Wireframe da Tela de Login. O layout é dividido em duas colunas principais. A coluna da esquerda contém um retângulo azul grande, destinado à logo. A coluna da direita contém o formulário de login. No topo da coluna da direita, há o texto "SEJA BEM-VINDO". Abaixo dele, há dois campos de entrada: "Usuário" e "Senha". Abaixo dos campos, há um botão verde com o texto "Entrar".

2) Wireframe da Tela de Cadastro

Wireframe da Tela de Cadastro. O layout é dividido em duas colunas principais. A coluna da esquerda contém o texto "CADASTRO". A coluna da direita contém o formulário de cadastro. No topo da coluna da direita, há o texto "Usuário". Abaixo dele, há um campo de entrada. Abaixo do campo, há o texto "Senha". Abaixo dele, há um campo de entrada. Abaixo do campo, há o texto "Confirmar Senha". Abaixo dele, há um campo de entrada. Abaixo dos campos, há um botão verde com o texto "CADASTRAR".

3) Wireframe do Dashboard:

Wireframe do Dashboard. O layout é dividido em duas colunas principais. A coluna da esquerda contém uma barra lateral azul com um círculo branco no topo e sete linhas de texto desordenadas. A coluna da direita contém o conteúdo principal. No topo da coluna da direita, há dois botões: "CADASTRAR NOVO JOGO" e "CADASTRAR PLATAFORMA". Abaixo dos botões, há uma tabela com 5 colunas: ID, JOGOS, PLATAFORMA, NOTA e AÇÕES. A tabela contém 1 linha de dados. Abaixo da tabela, há um espaço vazio.

ID	JOGOS	PLATAFORMA	NOTA	AÇÕES
1	Minecraft	Microsoft Store	5/5	○ ○ ○ ✨

4) Wireframe do CRUD do Jogo:

A wireframe for a form titled "CADASTRE UM NOVO JOGO". The form is centered on a purple background. It contains three input fields stacked vertically: "Nome do Jogo", "Categoria", and "Plataforma". Below these fields is a green button with the text "CADASTRAR".

5) Wireframe do CRUD da Plataforma:

A wireframe for a form titled "CADASTRE UMA NOVA PLATAFORMA". The form is centered on a purple background. It contains two input fields stacked vertically: "NOME DA PLATAFORMA" and "URL DA PLATAFORMA". Below these fields is a green button with the text "CADASTRAR".

Nos Wireframes acima, decidimos definir e apresentar:

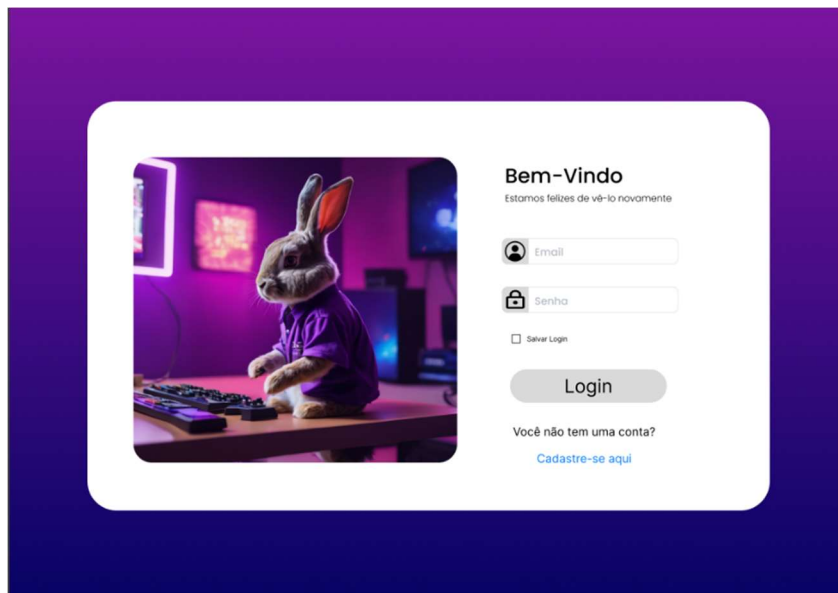
- Cores temáticas do site
- Estrutura do site
- Organização do site
- Traços do design, tentando o deixar intuitivo e minimalista.

Após isso, seguimos para o Figma para complementar o nosso rascunho e dar vida a nossa identidade.

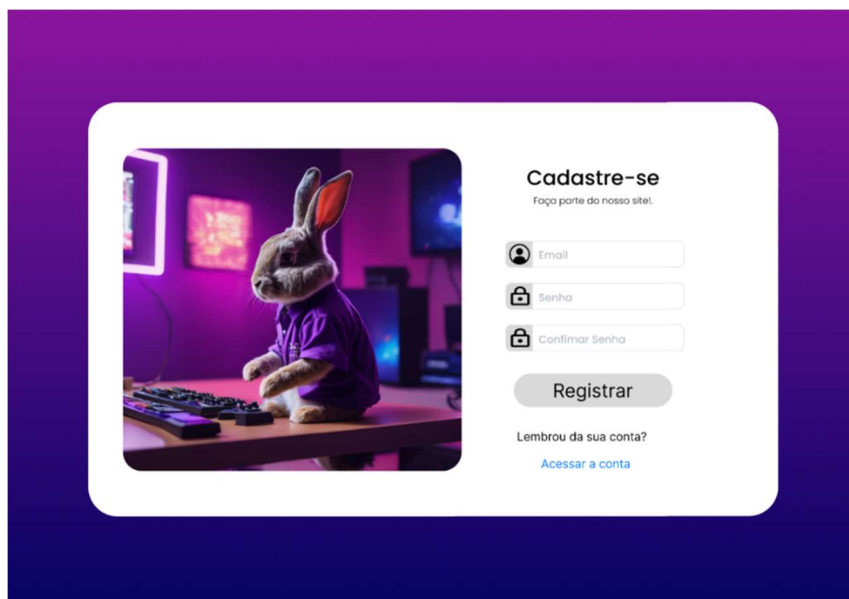
FIGMA

Abaixo está presente a tradução dos Wireframes para o Figma, dando início ao nosso processo de design e fortalecimento da identidade visual do nosso site.

1) Figma da Tela de Login:

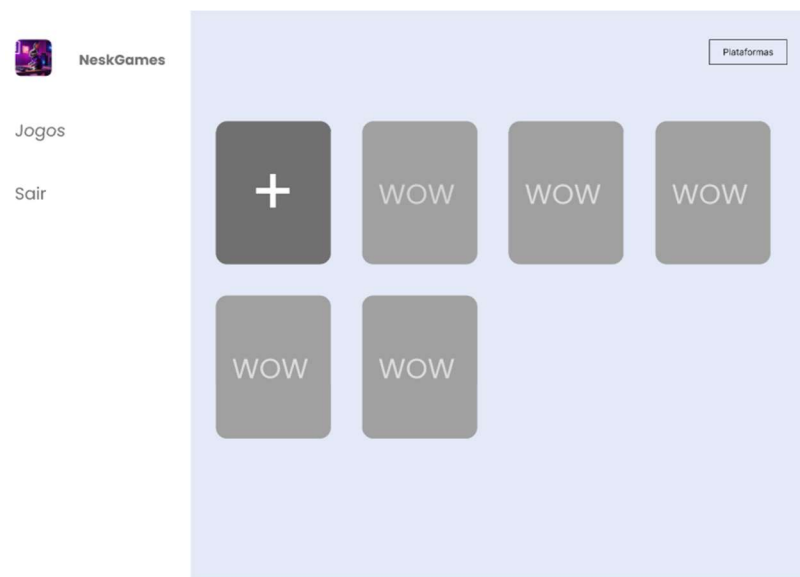


2) Figma da Tela de Cadastro:



3) Figma da Tela do Dashboard:

Obs: Inicialmente tivemos a ideia de transformar e formatar o dashboard em tabelas, porém futuramente mudamos para o formato de cards, como podemos ver abaixo.



4) Figma da Tela Inicial:



5) Figma da Tela de Cadastro de Jogos:

The form is titled "Cadastrar Jogo" and includes a close button "X" in the top right corner. It contains the following fields and options:

- Input field: "Nome do jogo"
- Input field: "Plataforma"
- Input field: "Upload da imagem"
- Radio buttons: "Recomendado" and "Não recomendado"
- Radio buttons: "Zerado" and "Não zerado"
- Range input: "Nota do jogo" with a slider and a value of "0"
- Submit button: "Cadastrar" (purple)

6) Figma da Tela de Edição de Jogo:

Editar Jogo

World of Warcraft

PC

Recomendado ☐ Não recomendado ☐

Zerado ☐ Não zerado ☐

Nota do jogo

10

deletar

Salvar

Nos protótipos de Figma acima, definimos:

- A identidade visual do site
- Estética do design
- Estilização do site

Após isso, demos início à programação e a construção real do site Nesk Games.

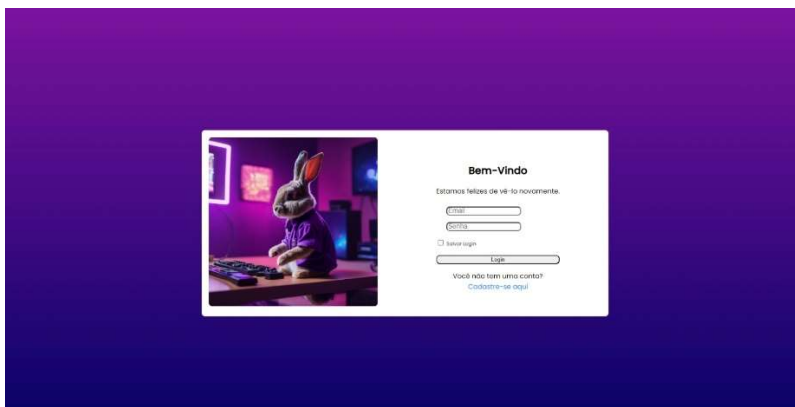
PROTÓTIPO EM HTML, CSS E JAVASCRIPT

Após todo o processo de elaboração de ideias, criação de rascunho e seus retoques, partimos para a criação do site com o HTML, CSS e Javascript.

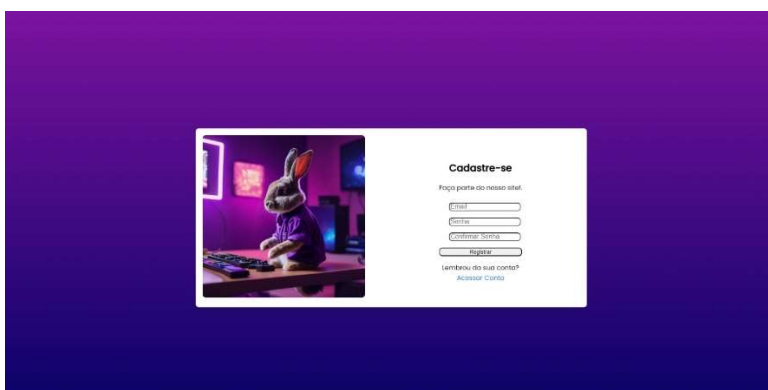
1) Protótipo da Página Inicial:



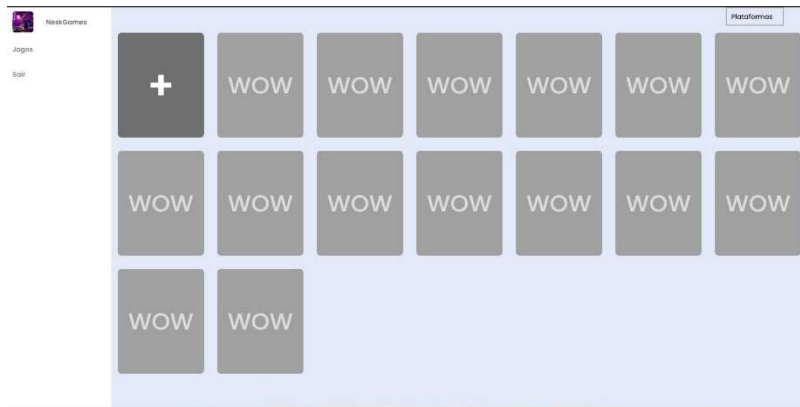
2) Protótipo da Tela de Login:



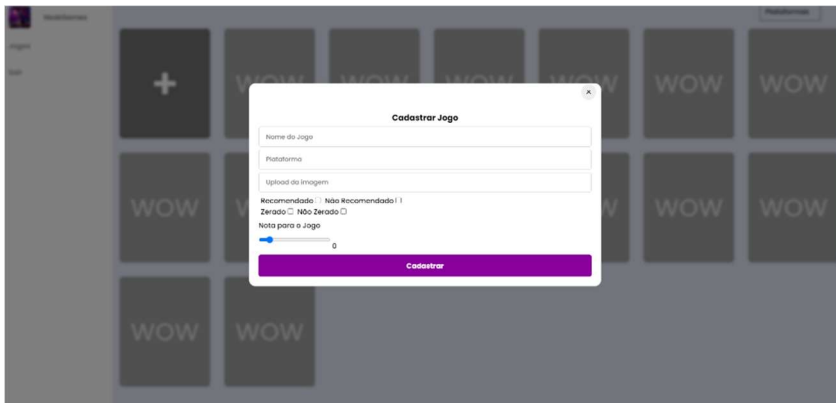
3) Protótipo da Tela de Cadastro:



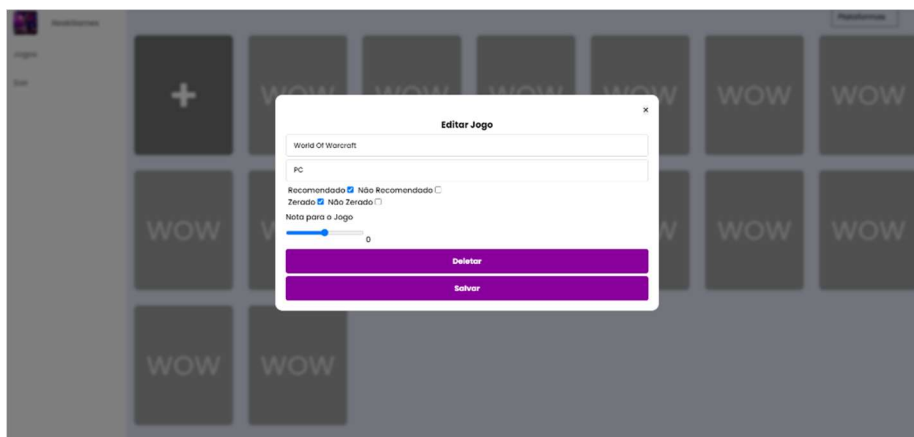
4) Protótipo do Dashboard:



5) Protótipo da Tela de Cadastro do Jogo:



6) Protótipo da Tela Edição de Jogos:



Por fim, nesse processo tivemos como referência umas metodologias adotadas por Nielsen, com a finalidade de melhorar a usabilidade do site.

Seguimos os princípios de Nielsen:

- **Estética e design minimalista:** uma plataforma precisa, sem informações desnecessárias para não poluirmos a tela e torná-la menos atrativa e interessante.
- **Consistência e padrões:** mantendo consistência e padrão visual (fonte, cor, desenhos, contornos, traços etc.) em todas as telas, sem desvirtuá-las

- **Liberdade de controle fácil para o usuário:** seguimos essa heurística principalmente na tela do Dashboard. Nos preocupamos em passar para o usuário a liberdade para ele conseguir fazer o que quiser dentro do sistema com os seus jogos.
- **Prevenção de erros:** construímos uma interface 100% intuitiva fazendo com que o usuário não se perca e saiba precisamente o caminho que deve ser feito para realizar os seus interesses dentro da plataforma.
- **Reconhecimento em vez de memorização:** projetamos o site para o usuário não ter a obrigação de decorar o caminho para chegar até as páginas realiza o que deseja. A interface é extremamente facilidade para que o usuário não precise se desgastar.

Essas foram as principais heurísticas levadas em conta para a criação e o desenvolvimento do projeto, levando-o para a versão final os mesmos princípios.

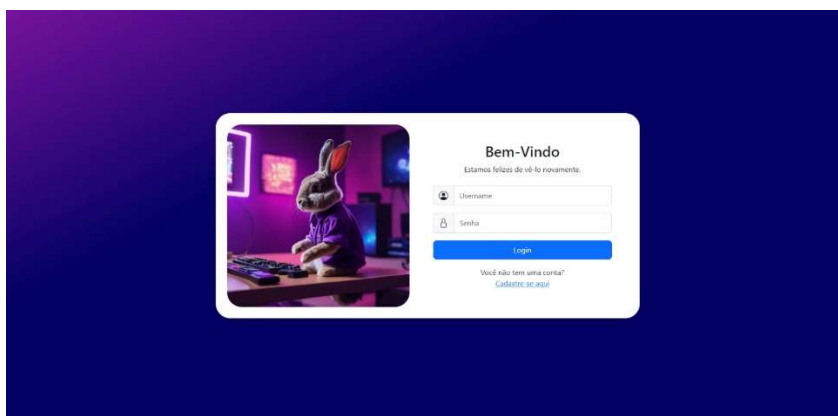
VERSÃO FINAL DA APLICAÇÃO WEB

Abaixo está todas as páginas do site em sua última versão, após toda a parte de programação e desenvolvimento.

1) Versão final da Página inicial:

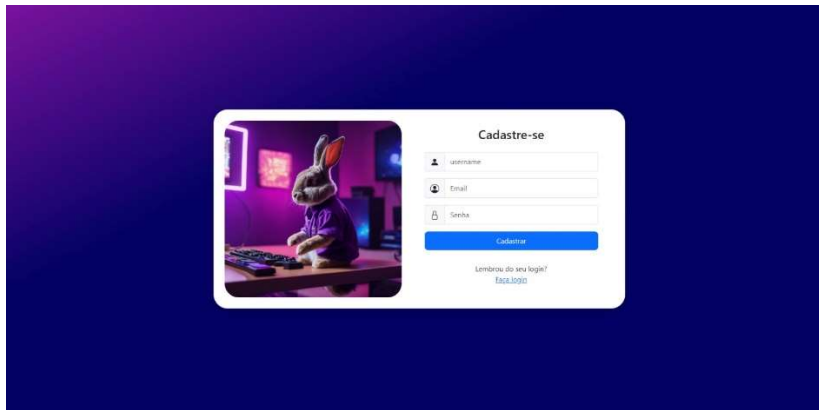


2) Versão final da Tela de Login:



3) Versão final da Tela de Cadastro:

Obs.: Assim que o usuário finalizar o seu cadastro, será redirecionada automaticamente para a tela de login



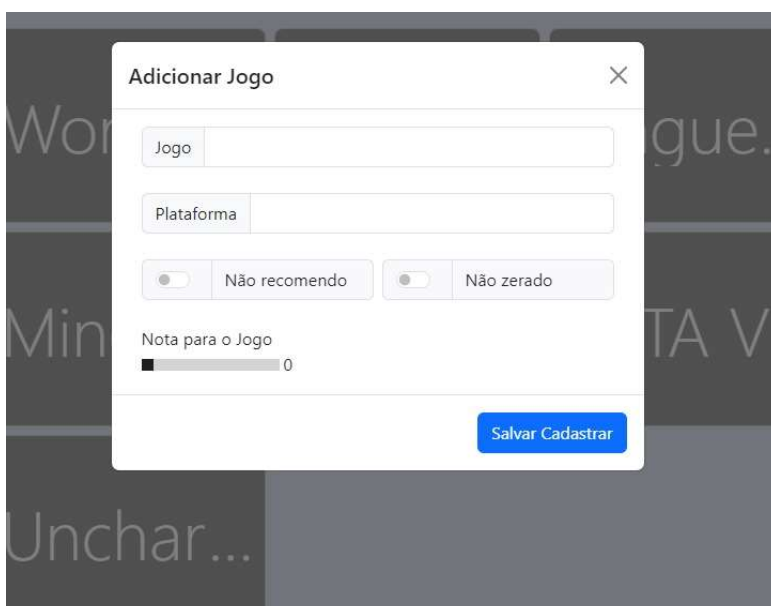
4) Versão final do Dashboard:



5) Versão final da Tela do Cadastro dos Jogos:

Obs.: o usuário poderá inserir:

- O nome do jogo
- Em que plataforma o jogo se encontra
- Se ele recomenda o jogo ou não
- Se o jogo já foi zerado por ele
- Qual a nota ele dará para o jogo



6) Versão final da Tela da Edição dos Jogos:

Obs.: Nesta aba, o usuário poderá editar todo o seu cadastro ou deletar, caso queira.



The image shows a modal window titled "Atualizar Jogo" with a close button (X) in the top right corner. Inside the modal, there are two input fields: "Jogo" with the value "Uncharted" and "Plataforma" with the value "Playstation". Below these are two toggle switches: "Recomendo" (turned on) and "Não zerado" (turned off). At the bottom left, there is a rating section labeled "Nota para o Jogo" with a progress bar and the number "3". At the bottom right, there are two buttons: "Deletar Jogo" (red) and "Salvar Cadastrar" (blue).

DETALHAMENTO DO CÓDIGO DO BACK-END E SUA INTEGRAÇÃO COM O FRONT-END

1. Detalhamento do código back-end:

Forma de inicialização do projeto:

- Primeiro passo (inicializando back-end):

Na pasta de back-end, executar *npm install* no terminal.

Executar *npm run dev* no terminal, para rodar o back-end de forma correta na porta 8080.

- Segundo passo (inicializando front-end):

Na pasta de front-end, executar *npm install* no terminal.

Executar *npm run start* no terminal, para rodar o back-end de forma correta na porta 8080.

Como foi feito o back-end:

Ao chegar no back-end, utilizamos NodeJS Express para criar a API RESTful para manipular solicitações HTTP e desenvolver o CRUD, além do manuseio de pacotes e dependências.

Para o banco de dados, foi utilizado o SQLite para armazenar e manipular dados em um arquivo local, sem a necessidade de um servidor dedicado.

Bibliotecas utilizadas:

- Express: utilizada para criar o servidor Web;
- Multer: Middleware para lidar com dados de formulário, permitindo a leitura de *FormData*;
- SQLite3: driver SQL que interage com o banco de dados;
- Cors: mecanismo que permite que uma aplicação web possa acessar recursos de outra origem diferente da sua.

Configuração Inicial:

- O servidor é configurado para escutar na porta 8080.
- Um objeto “upload” é criado usando Multer, para permitir a leitura de *FormData*.
- E um banco de dados é criado usando o módulo SQLite3

Endpoints:

Autenticação (/auth):

Endpoint POST para autenticar usuários com base em um banco de dados SQLite. Os detalhes do usuário são recebidos no corpo da solicitação, e a resposta indica se a autenticação foi bem-sucedida ou não.

Usuário (/user):

Endpoint GET para recuperar todos os usuários no banco de dados;

Endpoint PUT para adicionar um novo usuário ao banco de dados;

Endpoint POST para atualizar um usuário existente no banco de dados;

Endpoint DELETE para excluir um usuário do banco de dados.

Jogo (/game):

Endpoint GET para recuperar todos os jogos no banco de dados;

Endpoint PUT para adicionar um novo jogo ao banco de dados;

Endpoint POST para atualizar um jogo existente no banco de dados;

Endpoint DELETE para excluir um jogo do banco de dados.

Tratamento de Erros:

Em caso de erros durante operações no banco de dados, o servidor retorna mensagens de erro com status HTTP apropriados.

Instruções Finais:

O servidor é iniciado e escutado na porta 8080, e uma mensagem é exibida no console indicando o URL local para acessar o servidor.

2. Integração com o front-end

Como foi feita a integração:

Para efetuar a integração, utilizamos o Axios para conectar a API do back-end com o front-end.

O Axios é uma biblioteca Javascript utilizada para fazer requisições HTTP, oferecendo uma interface fácil e intuitiva.

Utilizamos uma *URL Reativa*, por meio do Axios, e criamos uma *URL default* para requisitar a API e facilitar a utilização de Routers

```
axios.defaults.baseURL = 'http://localhost:8080';
```

Veja que ele está solicitando o mesmo endereço onde está a API do back-end.