



# Informações do Projeto

---

**Título do Projeto**==Discô

**Curso**==Análise e Desenvolvimento de Sistemas

**Disciplina**==Trabalho Interdisciplinar: Aplicações Web

## 1. Navegador

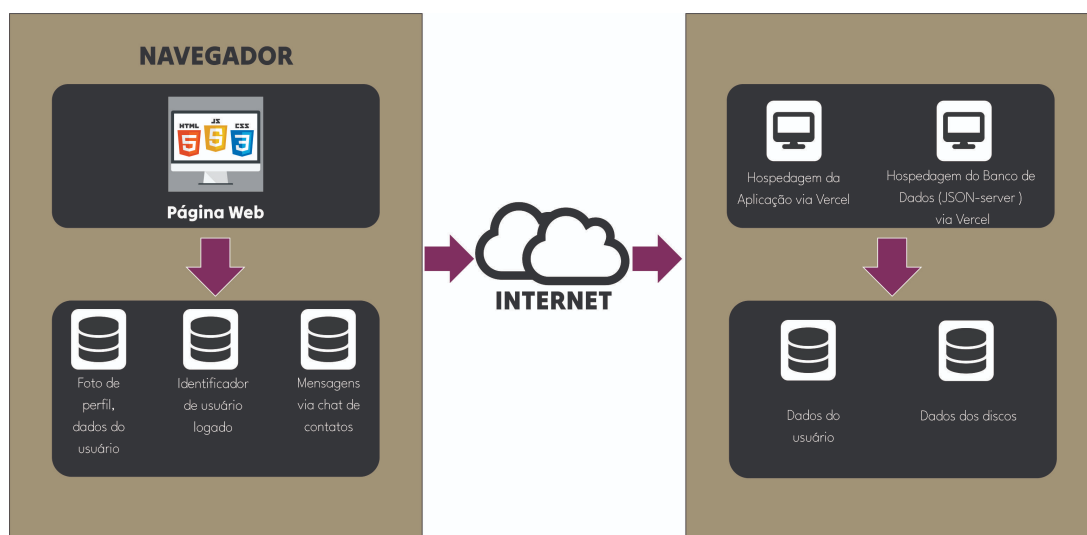
### Tecnologias Utilizadas

- HTML
- CSS
- JavaScript
- Framework Bootstrap
- - Bibliotecas HTML:
  - Font Awesome

### Funcionalidades

- **Interface do Usuário:**
  - Construída usando HTML, CSS e Bootstrap para um design responsivo.
- **Armazenamento Local:**
  - **Local Storage** para armazenar:
    - Fotos de perfil
    - Mensagens via chat

### Fluxo de Dados



- Os dados armazenados no Local Storage são utilizados para exibir informações no site.
- Interações do usuário (mensagens, atualizações de perfil, etc.) são armazenadas diretamente no Local Storage.

## 2. Nuvem

### Funcionalidades

### Tecnologias Utilizadas

- **Hospedagem:** Vercel
- **Banco de Dados:** API fake com json-server

### Funcionalidades

- **API REST:**
  - Servida pelo json-server, hospedado na Vercel.
  - Manipulação de dados do usuário, dados de venda e dados de avaliação.
- **Sincronização:**
  - O navegador se conecta com a API para buscar e atualizar dados conforme necessário.
  - Sincronização periódica entre Local Storage e a API para garantir que os dados locais e remotos estejam atualizados.

### Fluxo de Dados

- **Leitura de Dados:**
  - O navegador faz chamadas GET para a API no json-server para obter dados do usuário, vendas e avaliações.

### Estrutura de Dados:

```
{  
  "artista": "Milton Nascimento \n 'Clube da Esquina nº 2'",  
  "capa": "assets/img/MiltonNascimento.webp",  
  "genero": "MPB"  
},
```

```
"Usuarios": [  
  {  
    "name": "João Silva",  
    "email": "joao.silva@example.com",  
    "image": "img/fabiana.png",  
    "ranking": 5,  
    "socialMedia": "https://twitter.com/joaosilva",  
    "categories": [  
      "Rock",  
      "Jazz",  
      "Blues"  
    ]  
  }  
]
```

- **Escrita de Dados:**

- O navegador faz chamadas POST/PUT para a API para atualizar ou remover dados conforme necessário.
- Após operações de escrita, o Local Storage é atualizado para refletir as mudanças.

### 3. Integração

#### Comunicação Navegador-Nuvem

- **Fetch API:**

- Utilizada no navegador para fazer requisições assíncronas (GET, POST, PUT, DELETE) para a API no json-server.

#### Sincronização de Dados

- **Inicialização:**

- Ao carregar a página, os dados são primeiramente carregados do Local Storage.
- Caso não existam dados no Local Storage, uma requisição é feita à API para obter os dados iniciais.

- **Atualizações em Tempo Real:**

- Interações do usuário que modificam dados locais (como enviar uma mensagem ou atualizar o perfil) disparam requisições para atualizar os dados na API.
- Sincronização periódica para garantir que as alterações locais e remotas estejam em sincronia.

#### Estrutura de Requisições

- **GET /usuarios:** Obtém dados dos usuários.
- **POST /usuarios:** Cria um novo usuário.
- **PUT /usuarios/id:** Atualiza os dados de um usuário existente.

## Hospedagem

- Hospedar e lançar uma plataforma no Netlify envolve criar uma conta e conectar seu projeto ao serviço, garantindo que o domínio esteja devidamente configurado. Com cada atualização de código, o Netlify automaticamente realiza o deploy e reflete as mudanças. Adicionalmente, as ferramentas de gerenciamento e monitoramento do Netlify permitem uma análise contínua do desempenho do site.

[Site Discô](#)