



Frameworks WEB

ServerSide Frameworks

Preparando Banco de Dados



Índice

Introdução ao Server-side.....	3
Servidor estático.....	3
Servidor dinâmico.....	4
Single Page Application.....	4
Server-side Frameworks.....	5
Exemplos de Server-side Frameworks.....	5
Por que escolhemos o Express.....	5
Preparando o Banco de Dados.....	6
Usando MongoDB e Mongoose.....	6
Criando uma conta no MongoDB Atlas.....	7
Criando o banco de dados (Cluster).....	8
Security Quickstart.....	9
How would you like to authenticate your connection?.....	9
Where would you like to connect from?.....	9
Testando a conexão com o Banco de Dados.....	12
Constatando que os dados foram gravados com sucesso.....	12
Histórico de revisões.....	14



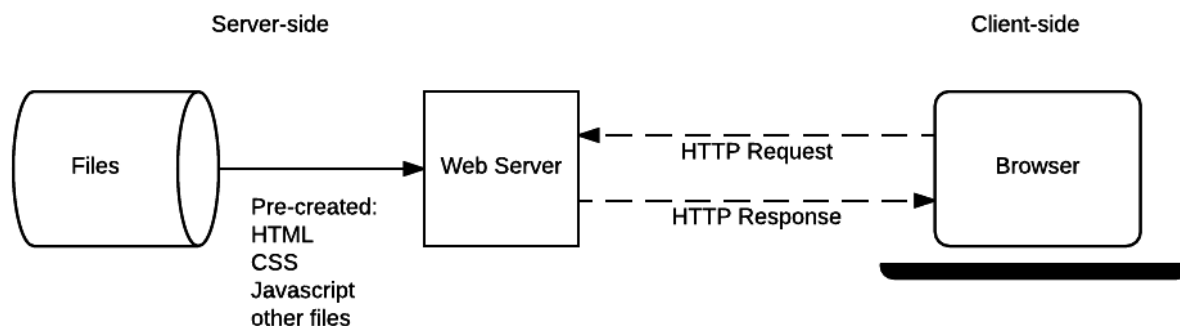
Introdução ao Server-side

Link do site de referência para esta parte do tutorial:

https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction

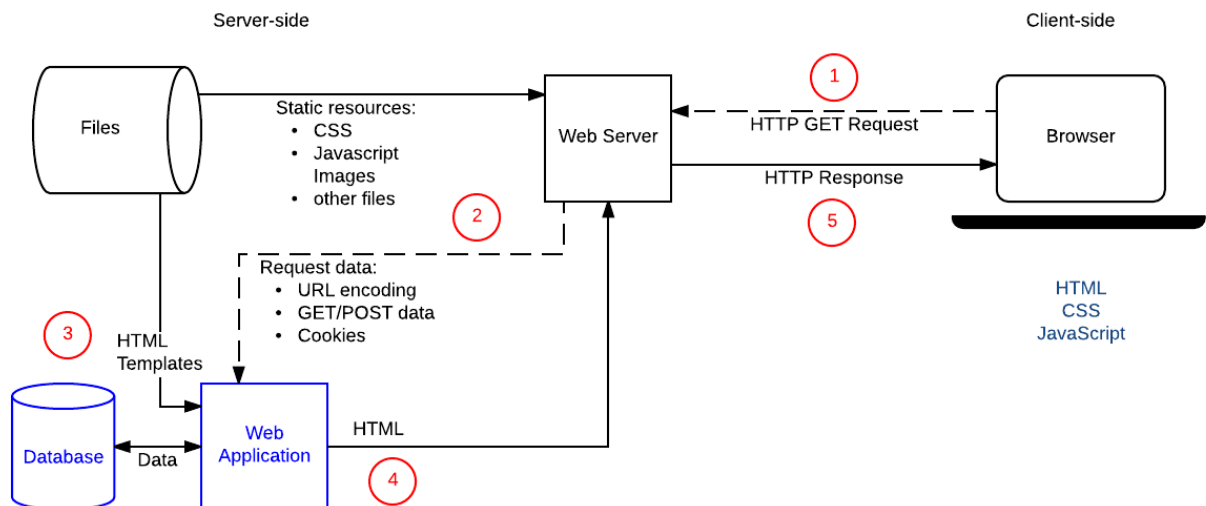
Servidor estático

Servidor envia documentos com conteúdo estático, previamente criado:

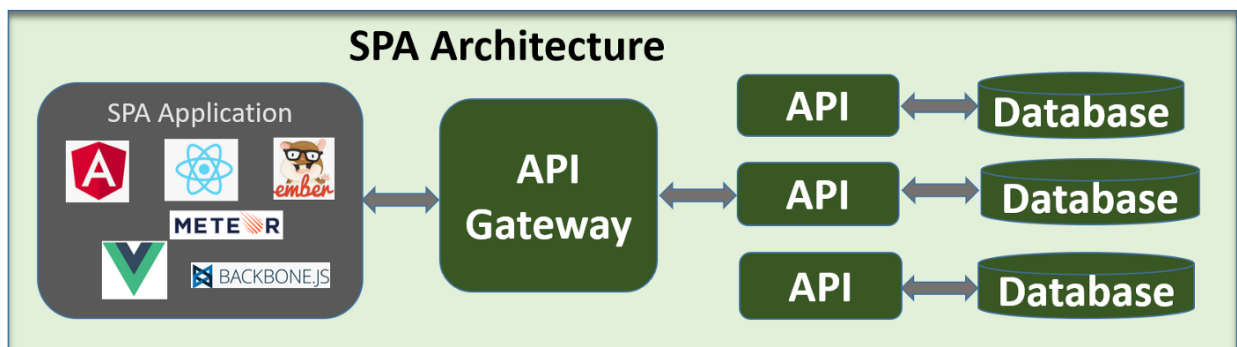




Servidor dinâmico



Single Page Application



<https://medium.com/nerd-for-tech/why-single-page-application-spa-architecture-is-so-popular-141b85400204>



Server-side Frameworks

Link do site de referência para esta parte do tutorial:

https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks

Exemplos de Server-side Frameworks

- Express (**JavaScript** - **Node.js**)
- Django (**Python**)
- Flask (**Python**)
- Deno (**JavaScript**)
- Ruby on Rails (**Ruby**)
- Laravel (**PHP**)
- ASP.NET (**Microsoft C#, Visual Basic, etc**)
- Mojolicious (**Perl**)
- Spring Boot (**Java**)

Por que escolhemos o Express

- Extremamente popular
- Rápido
- Minimalista
- Roda sobre o Node.js
- Roda JavaScript
- Faz parte de vários full stack frameworks: MEAN, KrakenJS, LoopBack, Sails etc



Preparando o Banco de Dados

Link do site de referência para esta parte do tutorial:

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose

Nesta parte do tutorial vamos entender como configurar um Banco de Dados para permitir que nosso app possa armazenar e buscar informações.

Um App Express podem utilizar diferentes ferramentas de bancos de dados (ex: PostgreSQL, MySQL, Redis, SQLite, MongoDB e etc), permitindo utilizar diferentes abordagens para a realização de CRUD (**C**reate, **R**ead, **U**ppdate and **D**eleete).

Usando MongoDB e Mongoose

MongoDB é um banco de dados open source NoSQL organizado em **Collections** e **Documents**, que funcionam de certo modo análogo a **Tabelas** e **Linhas** em banco de dados relacionais.

Mongoose funciona como uma interface para o MongoDB oferecendo a possibilidade de definição de Schemas e outras funcionalidades.

Como uma experiência adicional, ao contrário de utilizarmos um banco de dados local, vamos criar uma conta no **MongoDB Atlas** para utilizarmos um banco de dados hospedado na nuvem.



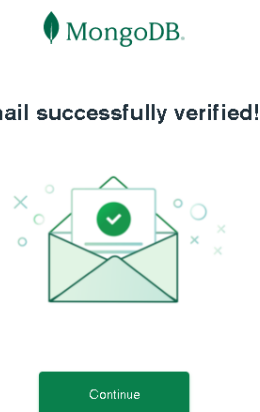
Criando uma conta no MongoDB Atlas

Acessar o seguinte endereço para se registrar no MongoDB Atlas:

<https://www.mongodb.com/pt-br/cloud/atlas/register>

The screenshot shows the MongoDB Atlas registration interface. At the top, it says 'mongoDB. Atlas' with the logo. Below that, it says 'Comece agora gratuitamente' and 'Não é necessário cartão de crédito'. There is a blue button with the Google logo and the text 'Entrar com o Google'. Below this, there is a section with the word 'ou' and several input fields: 'Sua empresa (opcional)', 'Seu e-mail de trabalho', 'Nome', 'Sobrenome', and 'Password'. The password field has a strength indicator. Below the fields, it says 'Mínimo de 8 caracteres' and 'Concordo com os termos de serviço e a política de privacidade.' with a checkbox. At the bottom, there is a green button that says 'Comece agora gratuitamente' and a link 'Já tenho uma conta? Entrar.'

Em seguida, realize a verificação solicitada por email:



Realize login em sua conta através do seguinte endereço:

<https://cloud.mongodb.com/user/login>



Criando o banco de dados (Cluster)

Clique no botão "Build a database" para criar seu primeiro cluster.



Create a database

Choose your cloud provider, region, and specs.

Build a Database

Once your database is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#).

Selecione a opção "FREE" (Shared), pressionando o botão "CREATE"

The screenshot shows the 'Create a database' interface with three main options:

- PREVIEW Serverless**: For serverless applications that aren't critical with variable traffic. Minimal configuration required. Features include: Pay only for the operations you run, Resources scale seamlessly to meet your workload, and Always-on security and backups. Starting at \$0.30/1M reads.
- ADVANCED Dedicated**: For production applications with sophisticated workload requirements. Advanced configuration controls. Features include: Network isolation and fine-grained access controls, On-demand performance advice, and Multi-region and multi-cloud options available. Starting at \$0.08/hr* (estimated cost \$66.94/month).
- FREE Shared**: For learning and exploring MongoDB in a cloud environment. Basic configuration options. Features include: No credit card required to start, Explore with sample datasets, and Upgrade to dedicated clusters for full functionality. Starting at **FREE**.

A red arrow points to the 'Create' button of the 'FREE Shared' option.

Certifique-se de manter selecionada a opção "M0 Sandbox" para Cluster Tier e clique no botão verde "Create Cluster".



Security Quickstart

Na sequência, vamos definir as opções de segurança para acesso futuro ao banco de dados, configurando:

- Usuário e senha para autenticar ao banco de dados
- Lista de endereços IP permitidos para conectar ao banco de dados

How would you like to authenticate your connection?

Selecione a opção **"Username and password"**.

Defina **"Username"** com: **joaoXXXXXX-mariaYYYYYY**, substituindo joaoXXXXXX e mariaYYYYYY pelos nomes e RAs dos integrantes do grupo.

Defina **"Password"** com **uma senha de sua preferência**

Clique no botão **"Create user"**

Where would you like to connect from?

Selecione a opção **"My local environment"**

Preencha o campo **"IP Address"** com: **0.0.0.0/0**

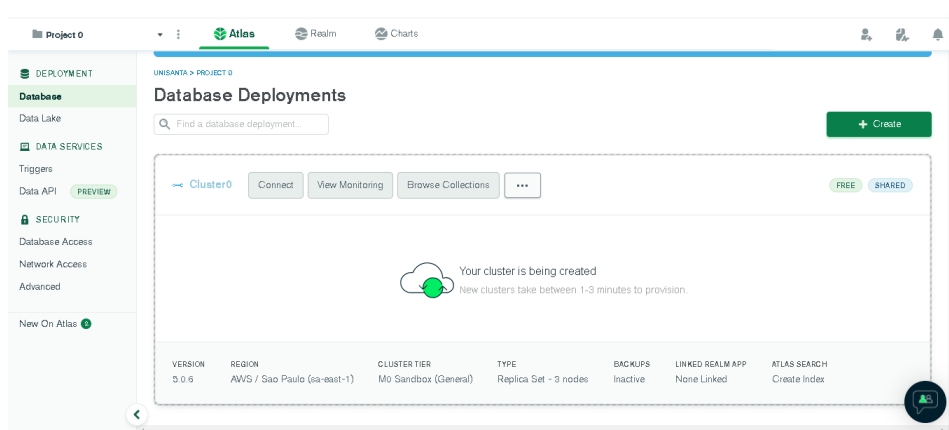
Ao definirmos o IP 0.0.0.0/0 permitiremos acesso a partir de qualquer endereço. Perceba que esta opção não é segura, porém facilitará nosso trabalho durante a elaboração do sistema de estudo.

Preencha o campo **"Description"** com: **Teste**

Pressione o botão **"Add Entry"**

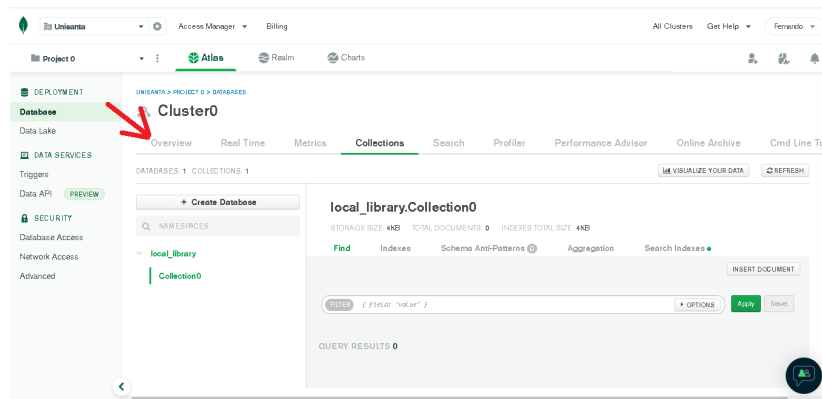
Pressione o botão verde **"Finish and Close"**

Aparecerá a seguinte tela, informando que seu cluster está sendo criado:

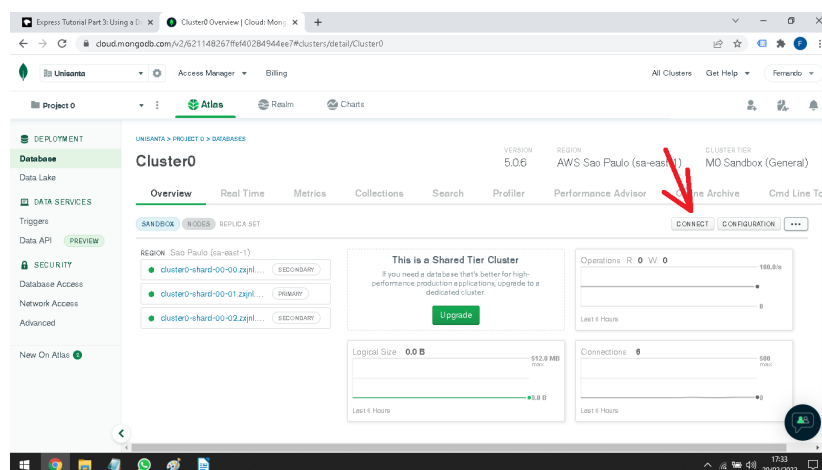




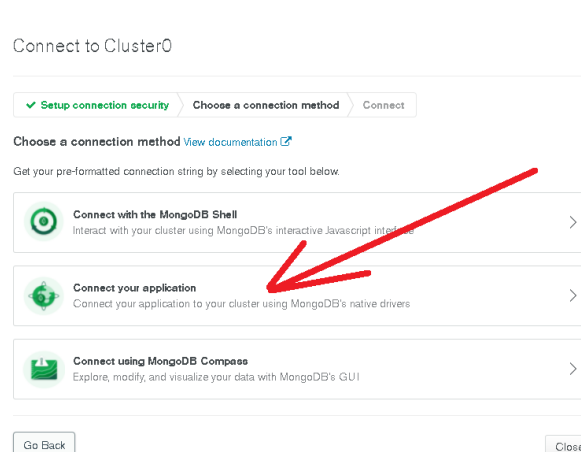
Clique em **"Overview"**:



Clique em **"CONNECT"**:



Clique em **"Connect your application"**:





Copiar a URL de conexão apresentada no destaque abaixo:

Connect to Cluster0

✓ Setup connection security

✓ Choose a connection method

Connect

1 Select your driver and version

DRIVER

Node.js

VERSION

4.0 or later

2 Add your connection string into your application code

☐ Include full driver code example

```
mongodb+srv://joao123456-maria456789:
<password>@cluster0.zxjnl.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **joao123456-maria456789** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

Go Back

Close

Substitua **<password>** pela sua senha

Substitua **"myFirstDatabase"** por **"tasksDB"**

Salve essa URL como sendo a sua URL de conexão ao banco de dados.



Testando a conexão com o Banco de Dados

Baixe o projeto "teste-banco-dados.zip" disponível em:

brangs.com/training/teste-banco-dados.zip

Descompacte o conteúdo do arquivo e instale as dependências através do comando:

```
npm install
```

Execute o script através do seguinte comando, utilizando a sua URL de conexão:

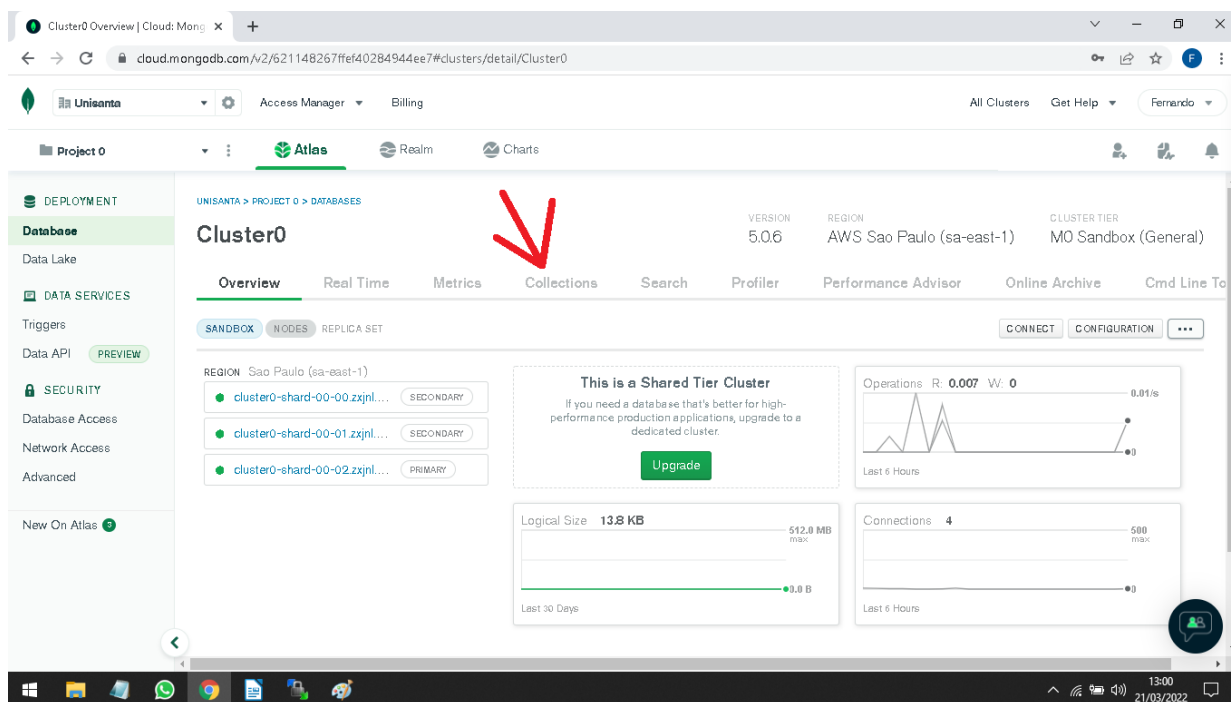
```
node populatedb "coloque sua URL de conexão aqui"
```

Obs: Coloque sua URL entre aspas

Constatando que os dados foram gravados com sucesso

Volte à sua conta no MongoDB Atlas

Cliente em "**Collections**"





Constata-se que 3 registros foram gravados com sucesso no banco de dados chamado **"tasksDB"** dentro da Collection chamada de **"tasks"**:

The screenshot displays the MongoDB Atlas web interface. On the left sidebar, the 'Database' section is expanded, showing 'tasksDB' and its collection 'tasks'. The main panel shows the 'tasksDB.tasks' collection with 3 documents. The 'QUERY RESULTS 1-3 OF 3' section shows the following documents:

Document
<pre>{ "_id": ObjectId("6239a1591b5a75b5fd96d7d9"), "description": "Eat" }</pre>
<pre>{ "_id": ObjectId("6239a1591b5a75b5fd96d7d9"), "description": "Eat" }</pre>
<pre>{ "_id": ObjectId("6239a1591b5a75b5fd96d7d9"), "description": "Eat" }</pre>



Histórico de revisões

Revisão: 00

Data: 21/03/2022

Descrição das alterações:
Documento original