

Next Level Week #2

📌 Aulas realizadas em 03/08/20 - 07/08/20

Links úteis:

- [Configuração do ambiente de desenvolvimento](#)
- [Layout do projeto](#)

1. Front-End #1

▼ Criando projeto

```
yarn create react-app web --template typescript
```

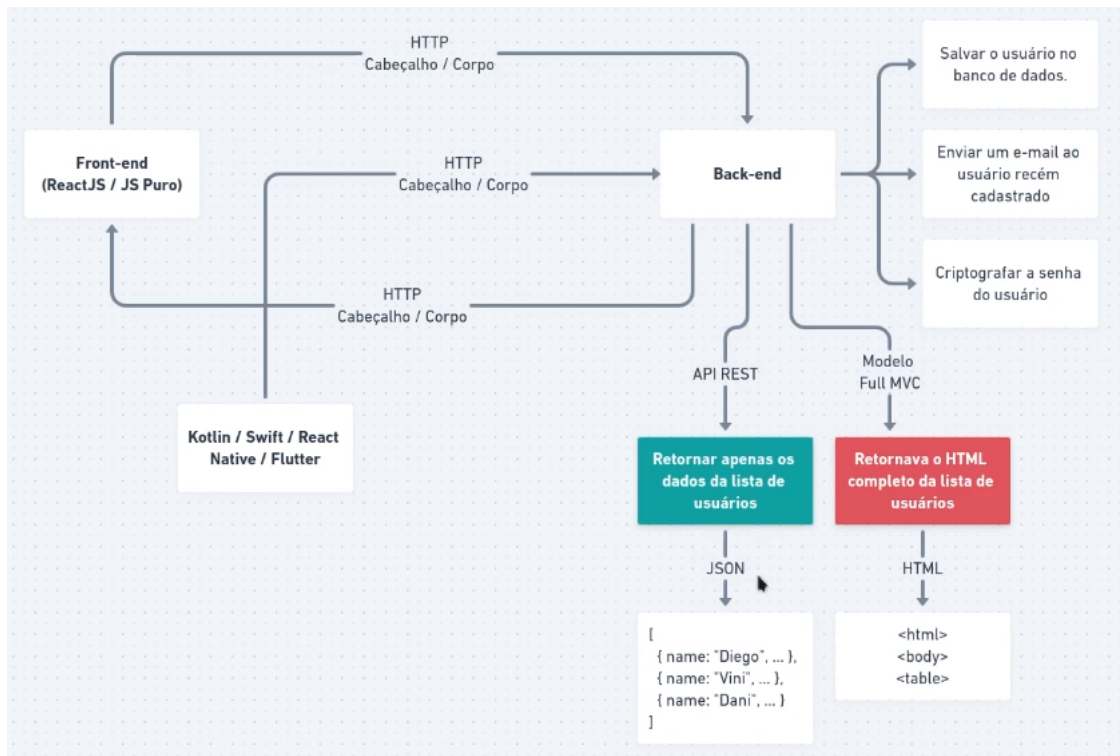
▼ Executando projeto

```
yarn start
```

2. Back-End

▼ MVC X API Restful

A **API Restful** permite que o back-end apenas retorne os **dados da aplicação**, geralmente em formato **JSON**, que pode ser **consumido** pelo **front-end** e pela aplicação **mobile**. Enquanto isso, o modelo MVC só retorna diretamente HTML, que não pode ser tão facilmente consumido por qualquer framework front-end ou pela aplicação mobile.



▼ Criando projeto

- Rodar o comando a seguir dentro da pasta **server**

```
yarn init -y
```

- Instalar o TypeScript

```
yarn add typescript -D
yarn tsc --init
```

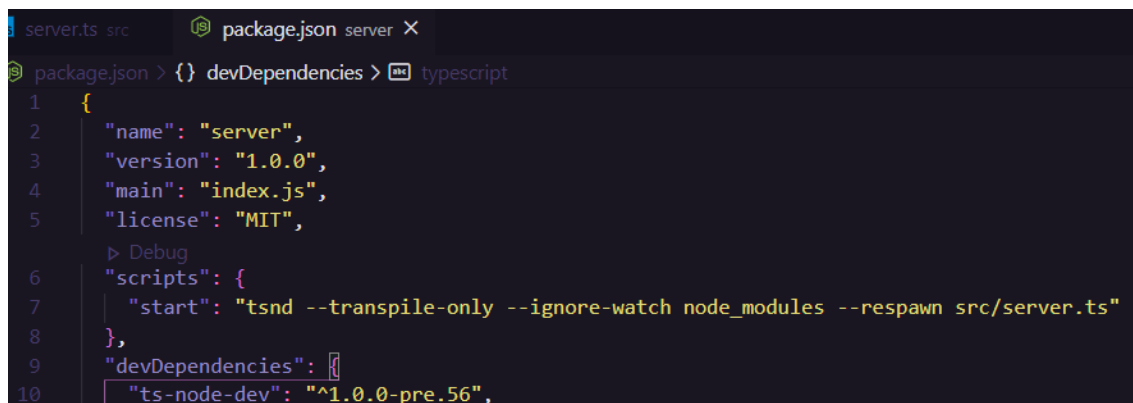
- Dentro de tsconfig.json, alterar o **target** para **"es2017"**

```
{
  "compilerOptions": {
    /* Visit https://aka.ms/tsconf
    /* Basic Options */
    // "incremental": true,
    "target": "es2017",
    "module": "commonjs",
    // "lib": [],
    // "allowJs": true,
```

- Para atualizar o servidor automaticamente

```
yarn add ts-node-dev -D
```

- Dentro de package.json, adicionar o **script** que permitirá rodar o projeto com `yarn start`



```
1 {
2   "name": "server",
3   "version": "1.0.0",
4   "main": "index.js",
5   "license": "MIT",
6   "scripts": {
7     "start": "tsnd --transpile-only --ignore-watch node_modules --respawn src/server.ts"
8   },
9   "devDependencies": {
10    "ts-node-dev": "^1.0.0-pre.56",
```

- `--transpile-only` para ele não procurar erros, só rodar o código
- `--ignore-watch node_modules` para ele não traduzir nada do node_modules
- `--respawn` para que sempre que acontecer uma alteração no código, o servidor reiniciar automaticamente
- Instalar o Express

```
yarn add express
yarn add @types/express -D
```

▼ Executando projeto

```
yarn start
```

▼ Identificando entidades e funcionalidades da aplicação

- **Conexões**
 - Rota para listar o total de conexões realizadas
 - Rota para criar uma nova conexão
- **Aulas**
 - Rota para criar uma aula
 - Rota para listar aulas
 - Filtrar por matéria, dia da semana e horário

▼ Banco de dados com Knex e SQLite

- Instalar o Knex

```
yarn add knex sqlite3
```

- Criar arquivos **database > connection.ts** e **knexfile.ts**

```
src > database > connection.ts > default
1  import knex from 'knex';
2  import path from 'path';
3
4  const db = knex({
5    client: 'sqlite3',
6    connection: {
7      filename: path.resolve(__dirname, 'database.sqlite')
8    },
9    useNullAsDefault: true,
10 });
11
12 export default db;
```

```
knexfile.ts > ...
1  import path from 'path';
2
3  module.exports = {
4    client: 'sqlite3',
5    connection: {
6      filename: path.resolve(__dirname, 'src', 'database', 'database.sqlite')
7    },
8    migrations: {
9      directory: path.resolve(__dirname, 'src', 'database', 'migrations')
10    },
11    useNullAsDefault: true,
12 };
```

- Criar as migrations dentro de **migrations**

```
EXPLORER
OPEN EDITORS
SERVER
> node_modules
v src
  database
    migrations
      00_create_users.ts U
      connection.ts U
      server.ts U
      .gitignore U
      knexfile.ts U
      package.json U
      tsconfig.json U
      yarn.lock U

00_create_users.ts migrations X
src > database > migrations > 00_create_users.ts > down
1  import Knex from 'knex';
2
3  // quais alterações queremos fazer no banco
4  export async function up(knex: Knex) {
5    return knex.schema.createTable('users', table => {
6      table.increments('id').primary();
7      table.string('name').nullable();
8      table.string('avatar').nullable();
9      table.string('whatsapp').nullable();
10     table.string('bio').nullable();
11   });
12 }
13
14 // se der problema, o que deve ser desfeito
15 export async function down(knex: Knex) {
16   return knex.schema.dropTable('users');
17 }
```

```
src > database > migrations > 01_create_classes.ts > down
1  import Knex from 'knex';
2
3  // quais alterações queremos fazer no banco
4  export async function up(knex: Knex) {
5      return knex.schema.createTable('classes', table => {
6          table.increments('id').primary();
7          table.string('subject').nullable();
8          table.decimal('cost').nullable();
9
10         table.integer('user_id')
11             .nullable()
12             .references('id')
13             .inTable('users')
14             .onUpdate('CASCADE')
15             .onDelete('CASCADE');
16     });
17 }
18
19 // se der problema, o que deve ser desfeito
20 export async function down(knex: Knex) {
21     return knex.schema.dropTable('classes');
22 }
```

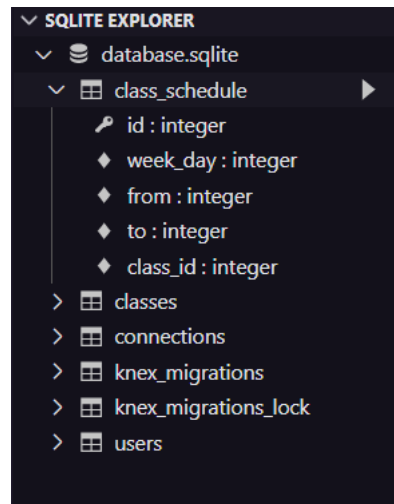
```
src > database > migrations > 02_create_class_schedule.ts > down
1  import Knex from 'knex';
2
3  // quais alterações queremos fazer no banco
4  export async function up(knex: Knex) {
5      return knex.schema.createTable('class_schedule', table => {
6          table.increments('id').primary();
7
8          table.integer('week_day').nullable();
9          table.integer('from').nullable();
10         table.integer('to').nullable();
11
12         table.integer('class_id')
13             .nullable()
14             .references('id')
15             .inTable('classes')
16             .onUpdate('CASCADE')
17             .onDelete('CASCADE');
18     });
19 }
20
21 // se der problema, o que deve ser desfeito
22 export async function down(knex: Knex) {
23     return knex.schema.dropTable('class_schedule');
24 }
```

```
03_create_connections.ts migrations X
src > database > migrations > 03_create_connections.ts > down
1  import Knex from 'knex';
2
3  // quais alterações queremos fazer no banco
4  export async function up(knex: Knex) {
5      return knex.schema.createTable('connections', table => {
6          table.increments('id').primary();
7
8          table.integer('user_id')
9              .nullable()
10             .references('id')
11             .inTable('users')
12             .onUpdate('CASCADE')
13             .onDelete('CASCADE');
14
15             table.timestamp('created_at')
16                 .defaultTo('now()')
17                 .nullable();
18         });
19     }
20
21     // se der problema, o que deve ser desfeito
22     export async function down(knex: Knex) {
23         return knex.schema.dropTable('connections');
24     }
}
```

- Criar os scripts em package.json para rodar as migrations

```
00_create_users.ts migrations package.json server X
package.json > ...
1  {
2      "name": "server",
3      "version": "1.0.0",
4      "main": "index.js",
5      "license": "MIT",
6      "scripts": {
7          "start": "tsnd --transpile-only --ignore-watch node_modules --respawn src/server.ts",
8          "knex:migrate": "knex --knexfile knexfile.ts migrate:latest",
9          "knex:migrate:rollback": "knex --knexfile knexfile.ts migrate:rollback"
10     },
11     "devDependencies": {
12         "@types/express": "^4.17.7",
13         "ts-node-dev": "^1.0.0-pre.56",
14         "typescript": "^4.0.7"
15     }
16 }
```

- Rodar `yarn knex:migrate`
- Instalar a extensão **SQLite do VSCode**
- Clicar com o botão direito no `database.sqlite` e depois em **"Open Database"**
- Em **SQLite Explorer** será possível visualizar as tabelas criadas



▼ Adicionando CORS

```
yarn add cors  
yarn add @types/cors -D
```

```
server.ts src X  
src > server.ts > ...  
You, a few seconds ago | 1 author (You)  
1 import express from 'express';  
2 import cors from 'cors';  
3  
4 import routes from './routes';  
5  
6 const app = express();  
7  
8 app.use(cors());  
9 app.use(express.json());  
10 app.use(routes);  
11
```

3. Front-End #2

▼ Instalando Axios

```
yarn add axios
```

▼ Redirecionando para WhatsApp a partir de um Front-End Web

<https://faq.whatsapp.com/general/chats/how-to-use-click-to-chat/>

4. Mobile

▼ Criando projeto

```
expo init mobile
```

▼ Executando projeto

```
yarn start
```

4. Mobile

▼ Instalando fontes

```
expo install expo-font @expo-google-fonts/archivo @expo-google-fonts/poppins
```

▼ Navegação

```
yarn add @react-navigation/native  
  
expo install react-native-gesture-handler react-native-reanimated react-native-screens react-native-safe-area-context @react-navigation/native  
  
yarn add @react-navigation/stack  
  
yarn add @react-navigation/bottom-tabs
```

▼ Instalando Axios

```
yarn add axios
```

▼ Usando Storage local para salvar Favoritos

```
expo install @react-native-community/async-storage
```

▼ Melhorias possíveis: usar o Picker para simular um Input Select

<https://reactnative.dev/docs/picker>

5. Próximos passos

Versão 2.0 Proffy.