Node.js Parte 1: Inovando com JavaScript no backend

- Esse curso capacita a:
  - Entender o funcionamento do JavaScript como linguagem de backend;
  - Aprender a desenvolver uma aplicação web completa em JavaScript com Node.js;
  - Diminuir a quantidade e dificuldade do código com Middlewares;
  - Ganhar produtividade utilizando módulos externos e o Node Package Manager;
  - Trabalhar com diferentes formatos de dados nas requisições;

Aulas:

1. **Começando com Node:**
   - O que é a plataforma Node;
   - O que é o npm;
   - Como criar um simples servidor HTTP utilizando o módulo HTTP do Node;
   - Como utilizar o npm para iniciar um projeto Node;
   - Utilidades do arquivo package.json;
   - Utilizar o npm para instalação de pacotes de dependências de um projeto Node;

2. **Express e templates com Marko:**
   - O que é o Express;
   - Criar rotas utilizando o Express;
   - Como exportar informações e funcionalidades de nossos módulos Node usando a sintaxe module.exports;
   - O que é o Nodemon e como utilizá-lo para agilizar o desenvolvimento;
   - Criar templates dinâmicos com MarkoJS;

3. **Persistência de dados:**
   - O que é o SQLite;
   - Como utilizar o SQLite numa aplicação Node;
   - O que é o padrão DAO;
   - O que são Promises;
   - Como desenvolver o padrão DAO utilizando uma abordagem voltada para callbacks e voltada para Promises;

4. **Manipulando a requisição com Middlewares:**
   - **Middlewares**: são trechos intermediários de código que são executados entre o envio da requisição e seu tratamento pela rota ativada;

     De modo geral, é composto por um padrão de URLs que são usadas para o ativar e um callback que recebe a requisição, a resposta e uma

função, normalmente chamada de next, que deve ser invocada para que o processamento da requisição siga em frente.

```
const express = require('express');
const app = express();

app.use('*', (req, res, next) => {

   // código do middleware.
});
```

Dado que dentro do middleware tudo que está antes da chamada da função next é executado antes da rota ativada e o que estiver após a chamada de next é executado após a rota;

- Como configurar rotas para o método POST do protocolo HTTP com o Express;
- O que são middlewares e para que servem;
- O que é o módulo body-parser e como ele ajuda a obter os dados vindos no corpo de uma requisição POST;
- Como utilizar o módulo body-parser criando um middleware que o utilize;

5. **Evoluindo a aplicação:**
   - Como criar uma rota para método DELETE do protocolo HTTP;
   - Como criar variáveis de URL para recuperar seus valores em rotas;
   - A trabalhar com AJAX em aplicações Node;
   - A utilizar arquivos estáticos utilizando o middleware **express.static()**;
   - Como criar uma rota para o método PUT do protocolo HTTP;
   - A sobrescrever o método HTTP com o módulo **method-override**;