

Painel / Meus Cursos / Graduação / ENGENHARIA DE SOFTWARE / 2023-1 / PROGRAMAÇÃO WEB / — Semana 15 / APS15 - Atividade Prática Supervisionada

Iniciado em	terça, 27 jun 2023, 13:44
Estado	Finalizada
Concluída em	terça, 27 jun 2023, 13:45
Tempo empregado	1 minuto 28 segundos
Notas	1,52/1,52
Avaliar	1,50 de um máximo de 1,50(100%)

Questão 1

Completo Atingiu 0,38 de 0,38

Qual é a principal função de um gerenciador de dependências em um projeto de desenvolvimento de software?

- ☐ a. Ele cria a interface do usuário para o projeto.
- ☐ b. Ele converte o código fonte do projeto em código de máquina.
- ☐ c. Ele gerencia o processo de deploy e hosting do projeto.
- ☐ d. Ele otimiza o código do projeto para melhor desempenho.
- ☒ e. Ele rastreia e instala todas as dependências necessárias para o projeto funcionar corretamente.



Questão 2

Completo Atingiu 0,38 de 0,38

Considere o código abaixo:

```
1 //Express Framework
2 const express = require('express');
3 const app = express();
4
5 //Import Controllers
6 const ItemController = require('./app/controllers/ItemController');
7 const itemController = new ItemController();
8
9 const ListController = require('./app/controllers/ListController');
10 const listController = new ListController();
11
12 //Define json
13 app.use(express.json());
14
15 //Routes
16 app.get('/', function(req, res) {
17   res.send('Bem-vindo!');
18 });
19
20 //lista
21 app.get('/list', (req, res) => listController.findAll(req, res));
22 app.post('/list', (req, res) => listController.create(req, res));
23 app.get('/list/:id', (req, res) => listController.findById(req, res));
24 app.put('/list/:id', (req, res) => listController.update(req, res));
25 app.delete('/list/:id', (req, res) => listController.delete(req, res));
26
27 //item
28 app.get('/item', (req, res) => itemController.findAll(req, res));
29 app.post('/item', (req, res) => itemController.create(req, res));
30 app.get('/item/:id', (req, res) => itemController.findById(req, res));
31 app.put('/item/:id', (req, res) => itemController.update(req, res));
32 app.delete('/item/:id', (req, res) => itemController.delete(req, res));
33
34 //Start server
35 app.listen(3000, function() {
36   console.log('Servidor rodando na porta 3000');
37 });
38
39
```



Indique qual alternativa descreva o arquivo.

- ☐ a. app/models/Item.js - Este arquivo define o modelo de Item, que descreve a estrutura de dados de um item.
- ☐ b. database.js - Este arquivo contém a configuração do banco de dados, como as credenciais de acesso ao banco de dados e opções de conexão.
- ☐ c. app/controllers/ItemController.js - Este arquivo contém o controller para itens, que contém a lógica de negócios relacionada a itens.
- ☒ d. index.js com rotas - Este arquivo é o ponto de entrada principal da aplicação, onde as rotas são definidas e a aplicação é iniciada.
- ☐ e. app/models/index.js - Este arquivo normalmente é responsável por configurar e exportar todos os modelos definidos em outros arquivos do diretório de modelos.

Questão 3

Completo Atingiu 0,38 de 0,38

Considere a seguinte situação: você está desenvolvendo um cliente para uma API RESTfull.

Quando você envia uma requisição GET para um endpoint específico, você recebe uma resposta com código de status HTTP 500.

O que isso significa e qual seria a melhor forma de lidar com essa situação?

- ☐ a. O servidor retornou um erro de "Too Many Requests". Você deve implementar uma estratégia de backoff e tentar a requisição novamente após um período de tempo.
- ☐ b. O servidor retornou um erro de "Not Found". Você deve verificar se o endpoint foi digitado corretamente.
- ☐ c. O servidor retornou um erro de "Unauthorized". Você deve verificar se as credenciais de autenticação foram incluídas corretamente na requisição.
- ☐ d. O servidor retornou um erro de "Bad Request". Você deve verificar se a requisição foi formatada corretamente.
- ☒ e. O servidor retornou um erro de "Internal Server Error". Você deve entrar em contato com o administrador para resolver o problema.

Questão 4

Completo Atingiu 0,38 de 0,38

O que acontece quando você executa o comando ``npm install express`` em um projeto Node.js?

- ☒ a. O express é baixado e colocado no diretório ``node_modules`` do projeto e a referência é adicionada à seção ``dependencies`` do ``package.json``.
- ☐ b. O express é baixado e colocado no diretório ``node_modules`` do projeto, mas nenhuma outra informação é adicionada ao ``package.json``.
- ☐ c. O express é baixado e colocado no diretório ``node_modules`` do projeto e a referência é adicionada à seção ``devDependencies`` do ``package.json``.
- ☐ d. Apenas a referência do express é adicionada à seção ``dependencies`` do ``package.json``, sem download para o diretório ``node_modules``.
- ☐ e. Apenas a referência do express é adicionada à seção ``devDependencies`` do ``package.json``, sem download para o diretório ``node_modules``.



◀ Atividade Pré-Aula - Semana 15

Seguir para...

Atividade Pratica - S15 - INDIVIDUAL ▶

...

