

Exercícios – P2 – Reposição da aula do dia 25/04/2025

1. O que é Node.js e quais são suas principais características?
2. Explique o modelo de I/O não bloqueante do Node.js e como ele difere de modelos tradicionais.
3. Descreva o que é o Event Loop no Node.js e como ele funciona.
4. Quais são as vantagens e desvantagens de utilizar Node.js em comparação com outras tecnologias de servidor?
5. Explique o conceito de "Single-threaded" no Node.js e como ele consegue lidar com múltiplas requisições.
6. O que é o V8 Engine e qual sua relação com o Node.js?
7. Explique o papel do libuv no ecossistema do Node.js.
8. Descreva a arquitetura interna do Node.js, incluindo seus principais componentes.
9. O que são Streams no Node.js e quais são os principais tipos?
10. Explique o conceito de Buffer no Node.js e quando ele deve ser utilizado.
11. Como o Node.js lida com operações assíncronas? Explique detalhadamente.
12. Quais são as diferentes formas de lidar com código assíncrono em Node.js?
13. Explique o conceito de Promise em Node.js e como implementá-lo.
14. O que é async/await e como essa funcionalidade melhora o código assíncrono?
15. Como realizar operações em paralelo no Node.js? Cite exemplos.
16. O que são callbacks no Node.js e por que são importantes?
17. Explique o problema conhecido como "Callback Hell" e como evitá-lo.
18. Como implementar um callback personalizado em Node.js?
19. Quais são as melhores práticas ao trabalhar com callbacks no Node.js?
20. Compare callbacks com Promises e async/await no contexto do Node.js.
21. Como funciona o tratamento de erros em Node.js?
22. Explique a diferença entre erros síncronos e assíncronos no Node.js e como tratá-los.
23. O que é o objeto Error no Node.js e quais são suas propriedades principais?
24. Como implementar um middleware de tratamento de erros em uma aplicação Express.js?
25. Quais são as melhores práticas para tratamento de exceções em aplicações Node.js?
26. O que são módulos no Node.js e como funcionam?
27. Explique a diferença entre require() e import/export no Node.js.
28. Descreva o sistema de módulos CommonJS e como ele é utilizado no Node.js.
29. O que é o objeto module.exports e como ele difere de exports?
30. Como funciona o sistema de cache de módulos no Node.js?
31. O que é o NPM e qual sua importância no ecossistema Node.js?
32. Explique a diferença entre dependências de produção e de desenvolvimento no package.json.
33. Como funciona o versionamento semântico (SemVer) no NPM?
34. Quais são os comandos essenciais do NPM e para que servem?
35. Explique o que é o arquivo package-lock.json e qual sua importância.
36. Quais são os passos para criar e publicar um módulo no NPM?
37. Como estruturar um módulo Node.js seguindo as melhores práticas?
38. Explique como implementar testes para um módulo Node.js.
39. Como documentar adequadamente um módulo Node.js?

40. Quais são as considerações de segurança ao desenvolver um módulo Node.js?
41. O que é o REPL no Node.js e como utilizá-lo?
42. Quais são os comandos especiais disponíveis no REPL do Node.js?
43. Como criar um REPL personalizado para sua aplicação Node.js?
44. Quais são os usos práticos do REPL durante o desenvolvimento?
45. Como o REPL pode ser útil para depuração de código Node.js?
46. Como o Node.js interage com o sistema de arquivos? Explique os módulos principais.
47. Quais são as diferenças entre operações síncronas e assíncronas de arquivo no Node.js?
48. Como implementar um sistema de upload de arquivos em uma aplicação Node.js?
49. Explique como trabalhar com streams para manipulação eficiente de arquivos grandes.
50. Quais são as melhores práticas de segurança ao trabalhar com arquivos em Node.js?
51. Como criar um servidor HTTP básico utilizando apenas os módulos nativos do Node.js?
52. Explique o conceito de middleware no contexto de aplicações web Node.js.
53. Quais são as melhores práticas de segurança para aplicações Node.js em produção?
54. Como implementar autenticação e autorização em uma API Node.js?
55. Explique como configurar HTTPS em uma aplicação Node.js.
56. Quais são os principais frameworks web para Node.js e suas características?
57. Compare Express.js, Koa.js e Hapi.js em termos de arquitetura e casos de uso.
58. O que é o Nest.js e como ele difere de outros frameworks Node.js?
59. Explique o conceito de ORM (Object-Relational Mapping) no contexto de Node.js.
60. Como implementar uma API RESTful utilizando Express.js?
61. O que são bancos de dados NoSQL e quais são os principais tipos utilizados com Node.js?
62. Como conectar uma aplicação Node.js ao MongoDB utilizando Mongoose?
63. Explique os conceitos de Schema e Model no Mongoose.
64. Como implementar operações CRUD básicas com MongoDB em uma aplicação Node.js?
65. Quais são as melhores práticas ao trabalhar com bancos de dados NoSQL em Node.js?