

```
1  Crie um fork desse repositório, faça os testes, responda as perguntas e depois
   submeta um pull request.
2  {criar fork após logar meu usuário no gitbub}
3  Os testes envolvendo código devem ser feitos preferencialmente em C# ou Java/Kotlin
   (Android).
4  Cada codificação deve estar em uma pasta com o nome que está entre perênteses nas
   questões.
5  As questões teóricas devem ser respondidas em um pdf que também será adicionado ao
   GIT.
6
7  O primeiro teste será pegar os códigos e dar um run. Tem que rodar de primeira.
8
9  Como você se atualiza tecnicamente?
10
11 -Assino newsletter (Macoratti, devmedia e microsoft msdn)
12 -Curto no facebook diversas páginas direcionadas a programação e tecnologia.
13 -Sempre estou conversando sobre tecnologia com amigos.
14 -Curto diversos canais de tecnologia no youtube
15 -Tento trabalhar em alguns projetos pessoais, sempre empregando novas tecnologias.
16 -Sou assinante de diversos cursos na Udemy (Android, React,Xcode .etc)
17
18 Crie uma função para calcular o n-ésimo elemento da Sequência de Fibonacci
   (fibonacci).
19
20 public static class BLLFibonacci
21 {
22     private static int nSequencia = 0;
23
24     public static string FibonacciInterativa(long nElemento)
25     {
26         DateTime dtTempoInicial = DateTime.Now;
27
28         long x = 1, y = 0, nResultado = 0;
29
30         for (nSequencia = 0; nSequencia < nElemento; nSequencia++)
31         {
32             nSequencia++;
33             nResultado = x + y;
34             y = x;
35             x = nResultado;
36         }
37
38         TimeSpan oDif = DateTime.Now - dtTempoInicial;
39
40         return string.Format(" Resultado: {0}, Elemento: {1}, Sequencia: {2},
   Duração: {3} ",
41             nResultado, nElemento, nSequencia, oDif.TotalMilliseconds);
42     }
43
44     public static string FibonacciRecurativa(long nElemento)
45     {
46         long nResultado = 0;
47
48         nSequencia = 0;
49
50         DateTime dtTempoInicial = DateTime.Now;
51
52         nResultado = Fibonacci_Recurativa(nElemento);
53
54         TimeSpan oDif = DateTime.Now - dtTempoInicial;
55
56         return string.Format(" Resultado: {0}, Elemento: {1}, Sequencia: {2},
   Duração: {3} ",
57             nResultado, nElemento, nSequencia, oDif.TotalMilliseconds);
58     }
59
60     private static long Fibonacci_Recurativa(long nElemento)
61     {
62         nSequencia++;
63
64         if (nElemento <= 1)
65         {
66             return 1;
```

```

67         }
68
69         return Fibonacci_Recursiva(nElemento - 1) +
           Fibonacci_Recursiva(nElemento - 2);
70     }
71 }

```

Disponível em:

<http://localhost:52485/Fibonacci/Recursiva/20>
<http://localhost:52485/Fibonacci/Interativa/20>

Qual solução é mais performática, iterativa ou recursiva? Por que?

Resultado Interativa: Resultado: 89, Elemento: 20, Sequencia: 20, Duração: 0,0335
 Resultado Recursiva: Resultado: 10946, Elemento: 20, Sequencia: 21891, Duração: 0,9222

A vantagem de usar métodos recursivos consiste na simplicidade de implementação de algoritmos que são por natureza recursivos.
 Mas em geral implementações recursivas são menos eficientes e podem ser convertidas para implementações iterativas.

o método recursivo vai ser mais lento que uma solução usando um loop pelo fato dele fazer diversas chamadas a ele mesmo, levando a enche a pilha de chamadas.

Opcional: Qual é o 5287º elemento da sequência?

O que significa SOLID?

S.O.L.I.D é um acrônimo que representa cinco princípios da programação orientada a objetos e design de código

Single Responsibility Principle (Princípio da Responsabilidade Única)
 Open/Closed Principle (Princípio do Aberto/Fechado)
 Liskov Substitution Principle (Princípio da Substituição de Liskov)
 Interface Segregation Principle (Princípio da Segregação de Interfaces)
 Dependency Inversion Principle (Princípio da Inversão de Dependências)

O que são design patterns?

Design Patterns ou padrões de projetos são soluções generalistas para problemas recorrentes durante o desenvolvimento de um software.
 Não se trata de um framework ou um código pronto, mas de uma definição de alto nível de como um problema comum pode ser solucionado.

Design patterns são modelos que já foram utilizados e testados anteriormente, portanto podem representar um bom ganho de produtividade para os desenvolvedores. Seu uso também contribui para a organização e manutenção de projetos, já que esses padrões se baseiam em baixo acoplamento entre as classes e padronização do código.

Quais são os tipos de design patterns?

Padrões de Criação

- Abstract Factory
- Object pool
- Builder
- Factory Method
- Prototype
- Singleton

Padrões Estruturais

- Private class data
- Adapter
- Bridge
- Composite

130 -Decorator
131 -Façade
132 -Business Delegate
133 -Flyweight
134 -Proxy
135
136 Padrões Comportamentais
137 -Chain of Responsibility
138 -Command
139 -Interpreter
140 -Iterator
141 -Mediator
142 -Memento
143 -Observer
144 -State
145 -Strategy
146 -Template Method
147 -Visitor
148
149
150
151
152 Com quais você está familiarizado? Qual é a função deles?
153
154 -Facade
155
156 Define uma interface que torna mais simples o uso do subsistema.
157 A complexidade de uma classe é oculta por meio de uma fachada
158
159
160 -FactoryMethod
161
162 ocultar detalhes sobre a criação de objetos, fornecendo uma interface para que isso
aconteça.
163 A classe que implementa a interface decidirá qual objeto será criado
164
165 -Singleton
166
167 Desejar que uma classe tenha no máximo uma instância. Esse padrão visa garantir essa
característica
168 provendo um ponto de acesso global à classe, ou seja, apenas um objeto de uma
determinada classe poderá existir,
169 independentemente do número de requisições que sejam feitas para criação de um novo
objeto.
170
171 Opcional: Qual é sua opinião quanto ao uso de design patterns?
172
173 A padronização contribui diretamente para a redução do acoplament, e facilidade na
manutenção e gerenciamento do código e fonte.
174
175
176
177
178 Qual foi o último livro técnico que você leu? Quando foi isso?
179
180 2020 - Vue.JS na pratica - Daniel schimitz
181 2020 - Exam ref 70-486 - William penberthy
182 2020 - ASP.NET core - Daniel Roth, Rick Anderson e Shaun Luttin
183
184 Observação: se já tivermos lido e você for chamado para uma entrevista, perguntas
poderão ser feitas a respeito do mesmo.
185
186
187
188
189
190 Cite 3 maneiras diferentes de implementar Dependency Inversion.
191
192 -Módulos de alto nível não devem depender de módulos de nível inferior.
193 Ambos devem depender de abstrações
194
195 -Abstrações não podem depender dos detalhes.
196

```
197 -Detalhes (implementações concretas) devem depender de abstrações.
198
199 O que são ORMs?
200
201 Técnica para aproximar o paradigma de desenvolvimento de aplicações orientadas a
202 objetos ao paradigma do banco de dados relacional.
203 O uso da técnica de mapeamento objeto-relacional é realizado através de um mapeador
204 objeto-relacional que geralmente é a biblioteca
205 ou framework que ajuda no mapeamento e uso do banco de dados.
206
207 Quais você conhece bem?
208
209 Entity Framework
210 Nhibernate
211 Dapper
212
213 Opcional: Cite pelo menos 2 vantagens e 2 desvantagens de seu uso.
214
215 vantagens:
216
217 Facilidade de implementação
218 Desempenho superior ao ADO.
219
220 Desvantagem:
221 Dependencia de componente de terceiro.
222 Pouco controle dos dados para tratamento.
223
224 O que são microserviços?
225
226 Um micro serviço é uma pequena porção de software que roda de maneira independente,
227 focada em um único e pequeno
228 conjunto de atividades dentro de um conjunto de serviços muito maior.
229
230 Conclusão: microserviços é uma versão mais enxuta de web services e web APIs.
231
232 Quais são suas vantagens e desvantagens?
233
234 Vantagem:
235
236 Baixo acoplamento
237 Facilidade em aplicar novas features e implantarmos em produção.
238
239
240 desvantagens:
241
242 Sem um propósito único, por exemplo, cada microservice pode terminar fazendo muitas
243 coisas, crescendo em uma arquitetura de vários serviços monolíticos.
244 Isso fará com que, além de não colher os benefícios completos de uma arquitetura de
245 microservices.
246
247 PROJETO*****
248 *****
249 Com a seguinte representação de produto (crud):
250
251 {
252
253     "sku": 43264,
254
255     "name": "Batata frita Ruffles Cebola & Salsa",
256
257     "inventory": {
258
259         "quantity": 15,
260
261         "warehouses": [
262
263             {
```

```

264         "locality": "SP",
265
266         "quantity": 12,
267
268         "type": "ECOMMERCE"
269     },
270
271     {
272
273         "locality": "MOEMA",
274
275         "quantity": 3,
276
277         "type": "PHYSICAL_STORE"
278     }
279 ]
280
281 },
282
283 "isMarketable": true
284 }

```

Crie endpoints para as seguintes ações:

Criação de produto onde o payload será o json informado acima (exceto as propriedades isMarketable e inventory.quantity)

Edição de produto por sku

Recuperação de produto por sku

Deleção de produto por sku

Resultado:

http://localhost:52485/Produto/Consulta/Edicao	{Put}
http://localhost:52485/Produto/Consulta/Remover/43264	{Delete}
http://localhost:52485/Produto/Consulta/Inclusao	{Post}
http://localhost:52485/Produto/Consulta/43264	{Get}

(Devido ao tempo de 4 horas não foi possível concluir essa etapa)

Requisitos:

Toda vez que um produto for recuperado por sku deverá ser calculado a propriedade: inventory.quantity

A propriedade inventory.quantity é a soma da quantity dos warehouses

Toda vez que um produto for recuperado por sku deverá ser calculado a propriedade: isMarketable

Um produto é marketable sempre que seu inventory.quantity for maior que 0

Caso um produto já existente em memória tente ser criado com o mesmo sku uma exceção deverá ser lançada

Dois produtos são considerados iguais se os seus skus forem iguais

Ao atualizar um produto, o antigo deve ser sobrescrito com o que esta sendo enviado na requisição

A requisição deve receber o sku e atualizar com o produto que tbm esta vindo na requisição

Não é necessário o uso de bancos de dados.

Testes são bem vindos.

Você não deve levar mais do que 4 horas para o teste todo.