

## **COGTIVEXPERIENCE**

A **COGTIVEXPERIENCE** proporciona tanto ao gestor quanto ao profissional uma rápida experiência do que ambos irão vivenciar no dia a dia de trabalho.

O gestor terá a oportunidade de conhecer com mais profundidade o estilo e capacidade criativa do profissional e suas preferências.

O profissional terá a oportunidade de experimentar um pouco do produto e terá uma amostra do tipo de demanda que a COGTIVE poderá lhe apresentar, além de demonstrar suas habilidades de construção de software.

Como trata-se de uma experiência, caso o profissional sinta necessidade de mais esclarecimentos ou mesmo queira discutir alternativas técnicas com o gestor, poderá solicitar ao recrutador e será atendido conforme disponibilidade de agenda. A gestão do tempo também será um fator a ser observado por ambos.

Que seja uma experiência relevante para todos e leve a decisões mais sábias e conscientes!

E continuem contando com o nosso apoio durante todo o processo.

GENTE & CULTURA

## COGTIVEXPERIENCE

### Objetivo

Este é um exercício de construção de software em que você poderá demonstrar a utilização de princípios, abstrações e padrões de projetos. Não é um teste e não existe uma resposta correta. Existem soluções que são mais aderentes, flexíveis e abertas a mudanças.

É esperado que o profissional utilize os recursos que julgar necessários, incluindo *Princípios de Programação Orientada a Objetos*, *SOLID* ou *Design Patterns*.

### O Problema

Indústria 4.0 é um conceito que engloba automação e tecnologia da informação aplicado à manufatura. Neste cenário, todos os tipos de máquinas e dispositivos interagem, se comunicam e aprendem uns com os outros. Além disso, grandes volumes de dados são gerados e enviados para a nuvem para serem processados por diversos algoritmos. A COGTIVE utiliza IoT e Tablets para buscar os dados no chão de fábrica e então analisá-los.

O principal dado colhido é o "**Apontamento**". O **apontamento** consiste em um **evento** com data/hora de **início**, data/hora de **fim** e uma composição de outros dados relativos a equipamento, lotes e quantidades.

Apontamentos →	08:00 - 09:00	09:00 - 10:00			11:00 - 12:30			13:00 - 16:00			16:30 - 18:00		
Horário →	08:00	09:00	10:00		11:00	12:00		13:00	14:00	15:00	16:00	17:00	

O problema apresentado consiste em ler uma lista de apontamentos e analisá-los conforme as atividades descritas abaixo

### Formato de Entrada

Os dados estão disponíveis em arquivo CSV conforme layout abaixo. Os campos disponíveis são "*IdApontamento*", "*DataInicio*", "*DataFim*", "*NumeroLote*", "*IdEvento*" e "*Quantidade*". O campo "*NumeroLote*" pode ser nulo dependendo do "*IdEvento*" da linha sendo processada.

#### Trecho de Exemplo:

```
4506;01/03/2018 06:00:00;01/03/2018 07:50:00;;9;6
4507;01/03/2018 07:50:00;01/03/2018 12:00:00;18020155;1;7
4508;01/03/2018 12:00:00;01/03/2018 13:00:00;;23;8
4509;01/03/2018 13:00:00;01/03/2018 14:00:00;18020155;1;9
4511;01/03/2018 14:00:00;01/03/2018 14:00:00;18020155;2;11
4514;01/03/2018 14:00:00;01/03/2018 22:00:00;;19;14
4515;02/03/2018 06:00:00;02/03/2018 07:25:00;;9;15
```

## 1. Funcionalidade Calcular GAPS

Gaps são períodos de tempo não preenchidos entre os apontamentos.

Apontamentos →	08:00 - 09:00		09:00 - 10:00		GAP		11:00 - 12:30		GAP		13:00 - 16:00					
Horário →	08:00		09:00		10:00		11:00		12:00		13:00		14:00		15:00	

O programa deverá calcular todos os Gaps entre o primeiro e último apontamento. Deverá apresentar: **quantidade de gaps encontrados** e **período total de duração de gaps** (soma da duração de todos os gaps encontrados no formato *horas:minutos:segundos*).

Exemplo de Saída em Console

```
Quantidade de GAPS: 00000
Período Total: 00:00:00
```

## 2. Funcionalidade Calcular Quantidades Produzidas

Os apontamentos que possuem campo "IdEvento" igual a 1 ou 2 são considerados "Apontamentos de Produção". O campo "Quantidade" possui a quantidade produzida. O campo "NumeroLote" possui o número do lote. Apenas estes tipos de evento utilizam "Quantidade" e "NumeroLote".

O programa deverá calcular a **quantidade total produzida**. (soma da quantidade de todos apontamentos com eventos igual a 1 ou 2). Deverá calcular também os **3 lotes com maior produção**.

Exemplo de Saída em Console

```
Quantidade Total Produzida: 00000000
1º Lote 00000000 produziu 00000000
2º Lote 00000000 produziu 00000000
3º Lote 00000000 produziu 00000000
```

## 3. Funcionalidade Calcular Horas de Manutenção

Os apontamentos que possuem campo "IdEvento" igual a 19 são considerados "Apontamentos de Manutenção".

O programa deverá calcular **período total de duração das manutenções** (soma da duração de todos os eventos de manutenção no formato *horas:minutos:segundos*).

Exemplo de Saída em Console

```
Período Total De Manutenção: 00:00:00
```

## **Requisitos Obrigatórios**

O programa deve implementar uma classe base “**Apontamento**” para armazenar as entidades lidas do arquivo.

Os **Apontamentos de Produção** da funcionalidade 2 deverão ser implementados em uma subclasse especializada “**ApontamentoProducao**”.

## **Orientações**

- O programa deve ser codificado em .NET C#
- O programa pode ser codificado em qualquer IDE
- A solução pode ser entregue em um arquivo zipado ou poderá ser enviado link de um repositório no Git.