# Introdução

Este documento é parte do processo seletivo da empresa para o Time de desenvolvimento.

Antes de anunciar os requisitos é importante lembrar que você será avaliado segundo o que for entregue, logo tente demonstrar todo seu conhecimento, neste teste não é vergonha você preparar uma ogiva nuclear para matar uma formiguinha queremos saber até onde você é capaz de ir desenvolvendo um código em C# com .Net.

Esperamos que nosso time seja composto por pessoas que buscam sempre a excelência no que fazem, e por causa disso busquem entregar sempre o melhor. Por este motivo olhando seu teste esperamos:

1. Capricho: Uma boa organização, itens bem escritos, indentados, formatados. Uma boa apresentação do que se entrega é sim importante.
2. Boa capacidade de comunicação escrita: Saber documentar e informar por escrito é muito importante em uma comunicação de longo prazo.
3. Boa capacidade abstração lógica: Numa linguagem alto nível como C# o emprego de boas práticas de desenvolvimento de software exige conhecimentos de orientação a objeto.
4. Boa capacidade de organização: Num cenário de trabalho remoto a capacidade de se organizar e demonstrar de forma clara seu progresso assim como o que foi feito é necessário.

Os.: lembre-se. É uma avaliação e queremos conhecer você o melhor possível.

# Requisitos não funcionais

* Seu teste deverá ser feito em .Net. A versão pode ser escolhida pelo executor.
* A parte de front-end deverá estar em ASP.Net MVC.
* Deverá haver emprego de testes unitários (Somente para Sênior).
* Deverá haver emprego de DDD (Somente para Sênior).
* Deverá haver emprego de Paralelismo (Para Pleno e Sênior).
* Os métodos deverão ser assíncronos.
* Componentes plugáveis (Somente para Sênior).
* Documentação de requisitos e escopo de execução (Somente para Sênior).

# Requisitos funcionais

## Introdução

Joyce Vieira é uma analista de Recursos Humanos e mensalmente ela recebe uma série de arquivos dos diferentes departamentos para executar o fechamento do ponto e emitir a ordem de pagamento.

Para emitir a ordem de pagamento Joyce precisa validar os dias trabalhados e também as horas trabalhadas em cada dia.

Nos arquivos recebidos, Joyce ao observar cada Pessoa valida os seguintes requisitos:

1. Dias trabalhados. O trabalho segue de segunda a sexta e é esperado que o funcionário trabalhe 8 horas por dia mais 1 hora de almoço.
2. Valor da hora da pessoa.
3. Os dias não trabalhados são descontados da pessoa.
4. Horas não trabalhadas são descontadas da pessoa.
5. Horas extras são pagas.
6. Dias extras são pagos.

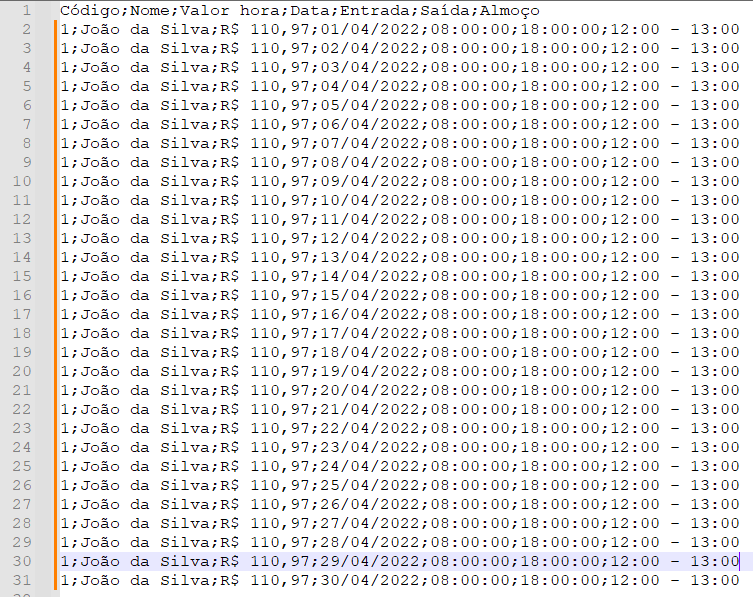
Após a análise de Joyce é gerado um arquivo contendo as seguintes informações:

1. O valor pago por departamento.
2. Os dados da pessoa.
3. Valor pago a cada pessoa.
4. Valor descontado de cada pessoa.
5. Quantidade de horas extras ou horas faltantes. Quando as horas são faltantes Joyce coloca o valor negativo.
6. Quantidade de dias extras ou faltantes. Quando os dias são faltantes Joyce coloca o valor negativo.

## Requisitos do sistema

O sistema pode ser um MVC ou um Console Application, mas ele deverá:

1. Ser feito em .Net (Qualquer versão).
2. Deverá conter um arquivo Readme com as informações do projeto.
3. Ler todos os arquivos da pasta designada. Exemplo: “C:\Arquivo a Importar\”.
4. A pasta deverá ser informada na hora em input feito pelo usuário.
5. Processar os arquivos em paralelo para aumentar o desempenho (Júniors não precisa fazer em paralelo). Joyce recebe em torno de 1000 arquivos com aproximadamente 10000 informações de usuário. (Sim, ela não dá conta de fazer tudo e por isso temos muito erro no RH - 😊)
   1. Se tiver uma solução melhor para resolver o problema de desempenho pode usar.
6. Os arquivos recebidos estão no formato CSV, Joyce tenta fazer o cálculo de tudo importando os arquivos no Excel e usando formulas que ela já conhece.
   1. Nome do arquivo contém: Nome do Departamento, Mês de vigência, Ano de vigência. Exemplo: ‘Departamento de Operações Especiais-Abril-2022.csv’
   2. O arquivo contém as seguintes colunas: Código: número, Nome: Texto, Valor hora: Dinheiro, Data: Dia do registro, Entrada: Hora do registro, Saída: Hora do registro, Almoço: Hora de registro



Essa imagem é ilustrativa, na realidade os dados não irão chegar de forma ordenada, mas serão formatados como neste exemplo.

Haverá mais funcionários e não há qualquer ordenação referente à data, código, nome ou qualquer outro campo.

Observe que as colunas serão separadas por ponto e vírgula.

Código;Nome;Valor hora;Data;Entrada;Saída;Almoço

1. O sistema deverá gerar uma saída no formato JSon. Um único arquivo para todos os departamentos. Será formatado como na imagem abaixo:



Para finalizar, pedimos que você crie um Projeto com ASP.Net MVC com uma Duas Views estáticas.

1. Home: Contendo o nome do candidato; Cargo pretendido; Pretensão salarial; Data do teste.
2. About: Gostaria que você colocasse um texto informando qual desafio de programação mais marcou sua jornada até o momento.

Neste texto queria que você colocasse: Tecnologia usada, Arquitetura, Como você atuou e por que te marcou tanto este desafio.