**Prova para Desenvolvedores .NET**

***Questão 1) Refactor***

**Detalhes da aplicação**

Para Maiores informações sobre o Teste, por favor acesse o link abaixo e siga as instruções que estão no documento.

<https://github.com/netshoes/net-lab/tree/master/TesteImposto/Misc>

**Dicas:**

* Cobrir o código de negócio com testes p/ aumentar a agilidade de evolução, além de trazer uma certa visão de impacto na evolução do código;
* código deve ser o mais simples e limpo possível, lembre-se que quase sempre nosso dia-a-dia é manutenção.
* Fique a vontade para refatorar o quanto você avaliar necessário.
* **Não efetue o Commit ou Pull Request neste repositório**

**Entrega:**

O Código pode ser entregue por e-mail (Zipado) ou em algum outro servidor de versionamento como Github, Bitbucket, etc.

Mande os links para o email do solicitante da prova.

***Questão 2) Dada uma stream, encontre o primeiro caracter Vogal, após uma consoante e que não se repita no resto da stream. O termino da leitura da stream deve ser garantido através do método hasNext(), ou seja, retorna falso para o termino da leitura da stream. Voce tera acesso a leitura da stream através dos métodos de interface fornecidos ao termino do enunciado.***

**\*Leia todo o enunciado**

**Premissas:**

Uma chamada para hasNext() ir retornar se a stream ainda contem caracteres para processar.

Uma chamada para getNext() ir retornar o proximo caracter a ser processado na stream.

Não será possível reiniciar o fluxo da leitura da stream.

Não poderá ser utilizado nenhum framework .NET, apenas código nativo.

**Exemplo:**

Input: aAbBABacfe

Output: e

No exemplo, ‘e’ é o primeiro caracter Vogal da stream que não se repete após a primeira Consoante.

Segue o exemplo da interface em C#:

public interface IStream{

public char getNext();

public boolean hasNext();

}

public static char firstChar(IStream input) {

}

**O que espera-se como resultado - Dicas e direcionamentos:**

* Tente criar sua implementação pensando em performance;
* Efetuar casos de teste para diversos cenários e uma boa pratica;
* Documentar o código seguindo as boas práticas de mercado;
* Os códigos mais simples tendem a serem mais elegantes;
* Em caso de não localização do caracter, o sistema deve informar uma mensagem amigável;
* Ao finalizar o desenvolvimento você pode deve enviar o código pelo Github/ Bitbucket ou envie por email do solicitante Netshoes. Se o arquivo for muito grande, em caso de arquivo compacto, envie o mesmo para um virtual drive e compartilha o link na prova;
* Caso utilize o Git/Bitbucket, não esqueça de criar o .gitignore
* Envie o link da resposta para o email do solicitando da prova;
* Não copie os códigos gerados na ferramenta.
* Não utilizar o nome da Netshoes nos projetos ou packages da prova.