**INSTRUÇÕES**

1. A partir de agora você tem **48 horas** para resolver o teste abaixo;
2. Crie uma documentação sobre todo o seu processo de desenvolvimento, deixando claro suas dificuldades e outros comentários que julgar importante;
3. Ao finalizar, ou no fim das **48 horas**, salve o documento e envie para os e-mails: [recrutamento@voxus.tv](mailto:recrutamento@voxus.tv) e [nathalia.ogura@voxus.tv](mailto:nathalia.ogura@voxus.tv);
4. Neste e-mail, você também deve colocar o **arquivo zipado do seu código,** e nele deve conter detalhes como utilizar aplicação / detalhes da aplicação, de forma geral.
   1. Caso não funcione enviar o arquivo zipado, pode salvar no Google Drive e nos enviar o link de acesso. :)
5. Para o exercício 2, utilize o arquivo excel “**Arquivo base - Desafio Voxus**”, fornecido no e-mail recebido junto com o teste.
6. O frontend do projeto não será levado em consideração.

**Teste - Estagiário Desenvolvedor Backend**

**ENUNCIADO**

1. You’re the developer responsible for creating our payment posting system on the Voxus web platform. This system should allow our financial team to create, update, delete, and list all payments directly through the platform.
   1. Create a web page with a function that allows users to insert new payments at the platform. The payment must contain a **title**, a **value**, a **date**, an **external tax** and comments.

A few rules must be applied before the record get inserted into the database:

* + Title must have between 5 and 100 characters;
  + Value must be a decimal value. **Any non numeric value must not be accepted**;
  + Date must be on the pattern Y-m-d (year-month-day);
  + External tax is a fixed value of **5%** of the **value** of the **payment** and must be automatically calculated by your code;
  + The observations box can or cannot be filled by the user;
  1. Create a web page with a function to list all the payments through the platform. All payment posting data should be listed, displaying **payment posting id,** **title**, **value**, **date**, **external tax** and **comments**.
  2. On the list of payments, create a function that deletes the selected payment on the click of a button.
  3. Still on the list of payments, create a function to edit payments that were already inserted in the platform. The same rules applied in the insertion should be considered on the editing feature.

1. Some payments are pre-saved on .xlsx files (a simple file from Excel) and our financial team needs to be capable of uploading these files into the platform. The platform should accept these files and save them into the database (use the .xlsx file “*Arquivo base - Desafio Voxus*”).
   1. Create a web page with a function that allows the user to upload a .xlsx file and keep the file saved in their project folder. No validation is needed, you just need the file uploaded and saved into the project folder.
   2. Change the behavior of the previous function to make your code read the uploaded file and create a new record in the database from the provided information on the file.

**Relembrando considerações importantes:**

* Para o exercício **2**, utilizar o arquivo excel fornecido no e-mail recebido junto com o teste.
* O frontend do projeto não será levado em consideração, então fique tranquilo!

Ah, não esquece de documentar todo o seu processo de desenvolvimento, deixando claro o que sentiu dificuldade ou facilidade na hora de desenvolver. Bom teste! :)