

CES - 29: Engenharia de Software

Prof. Inaldo

Integrantes Subgrupo: Lucas Nogueira, Daniel Lopes

Product Owner: Casd Vest (RH)

Scrum Master: Daniel Lopes

## Documentação Técnica - Projeto CASDVest



# 1- Considerações Iniciais

## 1.1- Objetivos do Projeto

No vestibular do CASDVest, ocorre um longo e burocrático processo que avalia as necessidades do aluno e família, a fim de escolher aqueles que mais se adequam com o perfil esperado para o ano. Neste processo, ocorre a entrevista de renda, na qual a assistente social preenche o perfil do convocado com os documentos de renda. Estes documentos - que são preenchidos manualmente e não são digitais – são levados no processo de tomada de decisão de quais alunos pleitearão uma vaga. O problema encontra-se no tempo e esforço utilizados principalmente na entrevista de renda. Por ser um processo manual e escrito, a assistente investe muito tempo. Além disso, o processo de escolha é bastante subjetivo, devido ao fato de que a assistente não possui uma ferramenta de exposição de dados que facilite a sua tomada de decisão. Portanto, objetiva-se, neste projeto, elaborar uma plataforma web que automatize e torne digital a extração dos dados dos alunos, bem como exponha-os de forma a auxiliar no processo de decisão.

## 2- Projeto

A documentação contendo o atual andamento do projeto encontra-se no seguinte link:

<https://trello.com/b/FSjHLroh/projeto-ces-29-casd-vest>

O projeto visa uma drástica redução de tempo e esforço para a execução da entrevista de renda, possuindo uma interface amigável e flexível, permite ao usuário consulta e atualização de diversos dados dos candidatos.

Além disso, o projeto já se encontra integrado com a atual plataforma web do Casd vest, permitindo não só uma interação com os funcionários mas também com os candidatos do vestibular. Os alunos podem consultar sua atual situação no processo seletivo e recebem e-mails de notificação sobre o vestibular.

Tais funcionalidades permitem uma facilidade de comunicação e maior transparência no processo organizado pelo Casd Vest. Vale ressaltar que o desenvolvimento envolve não somente o processo seletivo do Casd Vest mas também do Casdinho, voltado para estudantes do ensino médio.

## 3 - Guia de configuração do projeto

### 3.1 – Ferramentas de Configuração

#### 3.1.2 – XAMPP (Por quê e Como?)

O projeto foi desenvolvido em php, HTML e MySQL. Antes que o projeto finalizado seja totalmente integrado ao site do CASD Vest, pode-se abri-lo e testá-lo em localhost. Portanto, aqui serão abordados os requisitos e passos necessários para executar o projeto.

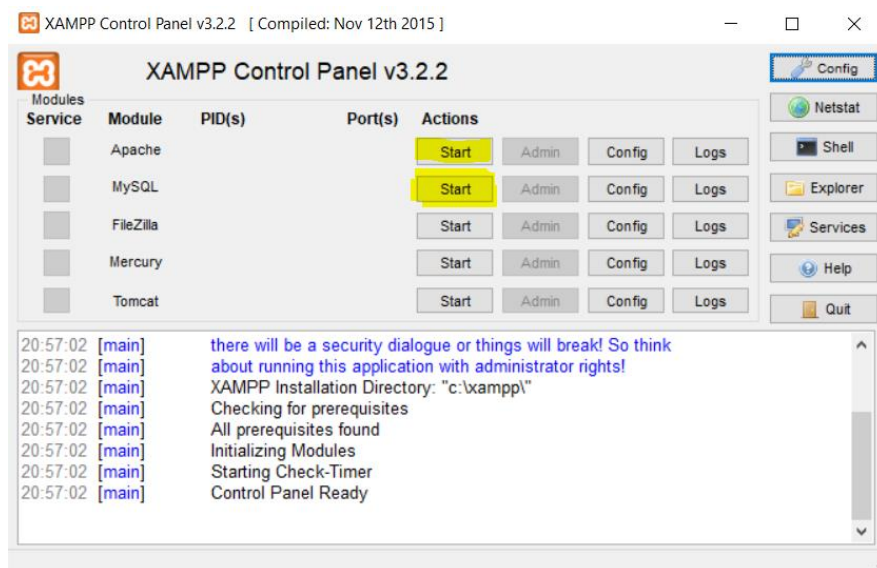
O ambiente local foi desenvolvido com a ferramenta XAMPP, fornecida pela Apache Friends. Ela foi escolhida por alguns motivos, mostrados abaixo, que a torna perfeita para o projeto.

- É a ferramenta mais usado para a criação de projetos PHP;
- É instalada sem o uso de linha de comando
- Contém uma interface gráfica simples e fácil de usar para inicializar seus módulos
- Contém diversos módulos, dentre eles, um servidor Apache (o servidor mais usado para PHP) e um servidor MySQL (database mais usado por projetos PHP, que será usado no projeto)
- Pode ser usado facilmente em ambientes Windows, Linux ou Mac. Em Linux é mais conhecido como LAMPP. O projeto abordará abordagens para instalação em Windows, mas um procedimento análogo ocorreria em Linux e Mac.

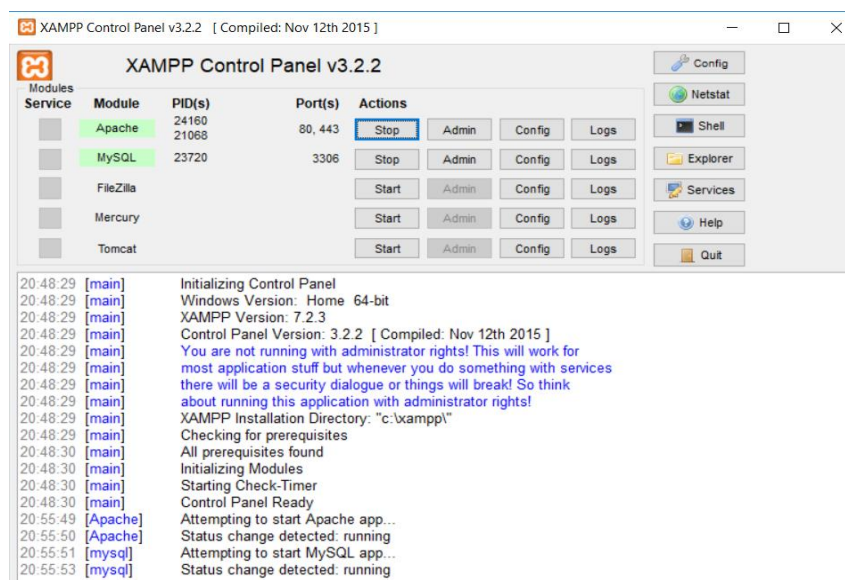


A inicialização do projeto com o uso da ferramenta XAMPP deve ser feito segundo os passos abaixo, que também são mostrados no arquivo README.md do Github.

1. Fazer o download do programa Xampp, que pode ser feito através do seguinte link: [https://www.apachefriends.org/pt\\_br/download.html](https://www.apachefriends.org/pt_br/download.html). Deve-se fazer o download da versão que mais se adequa a máquina que executará o projeto. Deve-se instalar o XAMPP em C:/xampp para evitar problemas de permissão.
2. Fazer o download da última versão do projeto em sua página do git, presente no seguinte link: <https://github.com/danirolopes/ces-29-casd-vest>.
3. Após isso deve-se colocar a pasta /santosdumont-inaldo do projeto em: C:\xampp\htdocs.
4. O próximo passo é abrir e configurar o Xampp, deve-se clicar no botão start dos dois primeiros itens: Apache e MySQL, conforme mostrado abaixo:



Após isso, os módulos serão acionados e deverão estar com a seguinte aparência:



5. Após isso, deve-se ir no browser e entrar em localhost/santosdumont-inaldo, resultando no seguinte resultado



Após isso, o programa estará funcionando em um servidor local normalmente.

Obs: A primeira requisição para o banco de dados pode demorar, pois ela vai criar o banco de dados automaticamente através de um autoloader.

### 3.1.2 – Composer (Por quê e Como?)

Composer é um gerenciador de dependências para projetos PHP, assim como o npm gerencia dependências de um projeto node.js. Ele foi escolhido pelos seguintes motivos:

- É o principal método de gerenciamento de dependências de um projeto PHP moderno. Sendo inclusive recomendado pelo próprio PHPUnit como método de instalação
- Cria dependências de maneira local no projeto, não de maneira global no computador. Assim, uma nova pessoa que vá instalar o projeto não tem que instalar o Composer, e sim rodar suas dependências já salvas e organizadas na pasta /vendor
- Não é preciso instalar o composer em novas instalações do projeto!!
- É facilmente instalado via linha de comando
- Consegue manusear as dependências do projeto através do arquivo composer.json

Para uma nova instalação do projeto, não é necessário instalar o composer, já que as dependências são salvas na pasta /vendor. Só é necessário instalar o composer caso se queira adicionar/remover dependências do projeto.

Caso seja necessário instalar o composer, siga os seguintes passos:

1. Baixar o executável do composer para Windows em <https://getcomposer.org/Composer-Setup.exe>
2. Cheque a versão com o comando “composer -V”
3. Baixe novas dependências via o comando “composer require”
4. Mude o arquivo composer.json para mudar as dependências do projeto. O composer reconhece as dependências automaticamente e as instala.

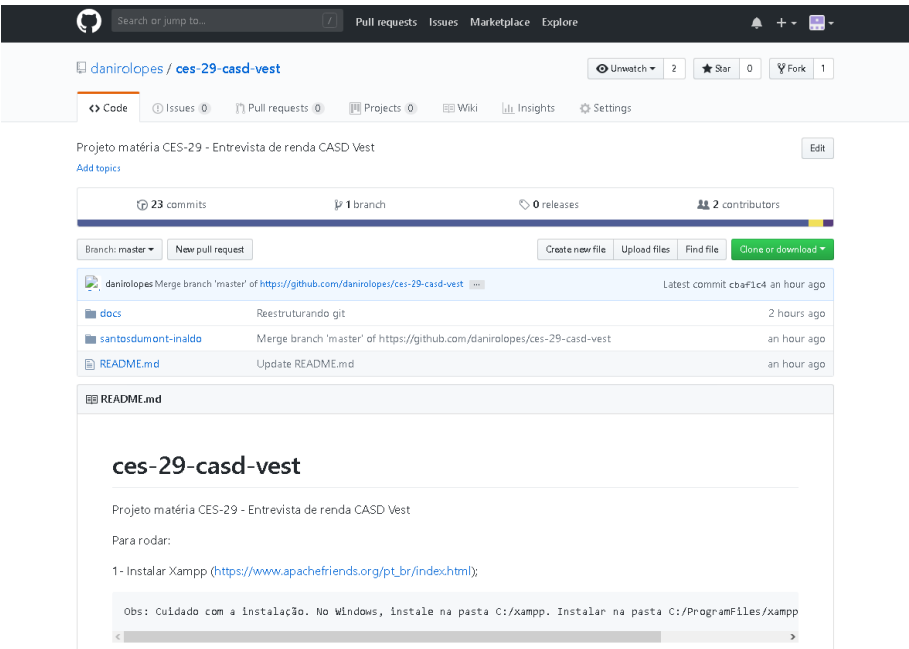
## 3 – Estrutura do GIT

O Git é uma ferramenta de desenvolvimento colaborativo de código. O Git do projeto está localizado em <https://github.com/danirolopes/ces-29-casd-vest>. A pasta raiz do Git contém 2 pastas e 1 arquivo, como mostra a figura abaixo.

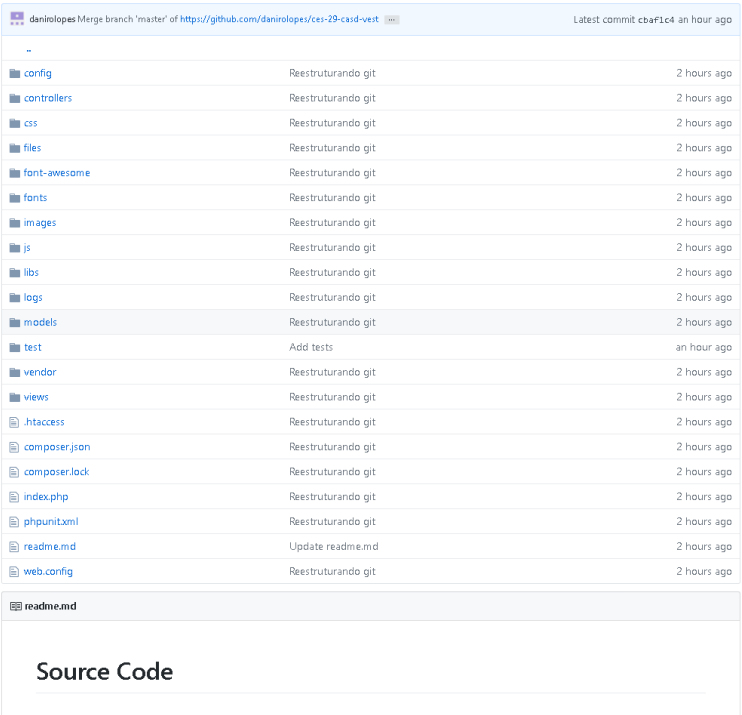
A pasta docs contém os arquivos de documentação. Desde relatórios produzidos pelo grupo, até gráficos de Burnup e BurnDown produzidos durante o projeto. Além disso, contém um README com o link do Trello utilizado pelo grupo.

A pasta santosdumont-inaldo contém de fato o source code do projeto, que deve ser usado durante sua instalação , conforme explicado acima.

E o arquivo README contém instruções para a configuração do código, para que possa ser executado por grupos futuros.



Dentro da pasta santosdumont-inaldo, que contém o source code, pode ser vista um conjunto de pastas e arquivos globais do projeto, como mostra a figura abaixo.



Cada uma das pastas tem sua função para a execução do projeto. Dentro dessa pasta santosdumont-inaldo, existe um arquivo README.md que explica qual a funcionalidade de cada pasta e de cada arquivo armazenado na root, informando quais podem ser modificados, quais não, e em quais situações eles podem ser modificados. O arquivo README.md pode ser visto na figura abaixo.

## Source Code

### Pastas:

```
/controllers - contém os controllers do projeto.  
  
/css - contém os arquivos de estilo .css do projeto.  
  
/files - contém os arquivos que serão usados para download e upload do projeto.  
  
/font-awesome  
  
/fonts  
  
/images  
  
/js - contém os arquivos de estilo .js do projeto.  
  
/libs - contém os plugins utilizados no projeto (swiftMailer e Bcrypt), além das classes pai Controller, Model etc.  
  
/logs - contém os logs php e pdo do projeto.  
  
/models - contém os models do projeto.  
  
/test - contém os arquivos de teste phpunit do projeto.  
  
/views - contém os arquivos de renderização .html do projeto.  
  
  
/vendor - contém as dependências do projeto (como o phpunit), pasta criada pelo composer  
  
  
/config - contém arquivos com variáveis de dependência global do projeto, como: PATHS, DATABASES e CONSTANTS. Também  
Altere somente aqui para mudar configuração, como nome da pasta do arquivo, nome de DB, e etc.
```

### Arquivos globais:

```
.htaccess e web.config - NÃO MEXER. Arquivos usado pelo Apache para sua configuração  
  
index.php - Arquivo que recebe a requisição e a redireciona, seguindo os padrões de um projeto MVP.  
Alterar a variável ENVIRONMENT caso se mude de localhost para production server  
  
composer.json - Arquivo usado pelo composer para gerenciar dependências do projeto. Só mexer caso queira se adicionar  
dependências  
  
phpunit.xml - Arquivo responsável por dizer onde os testes estão (No projeto, estão na pasta /test).
```