

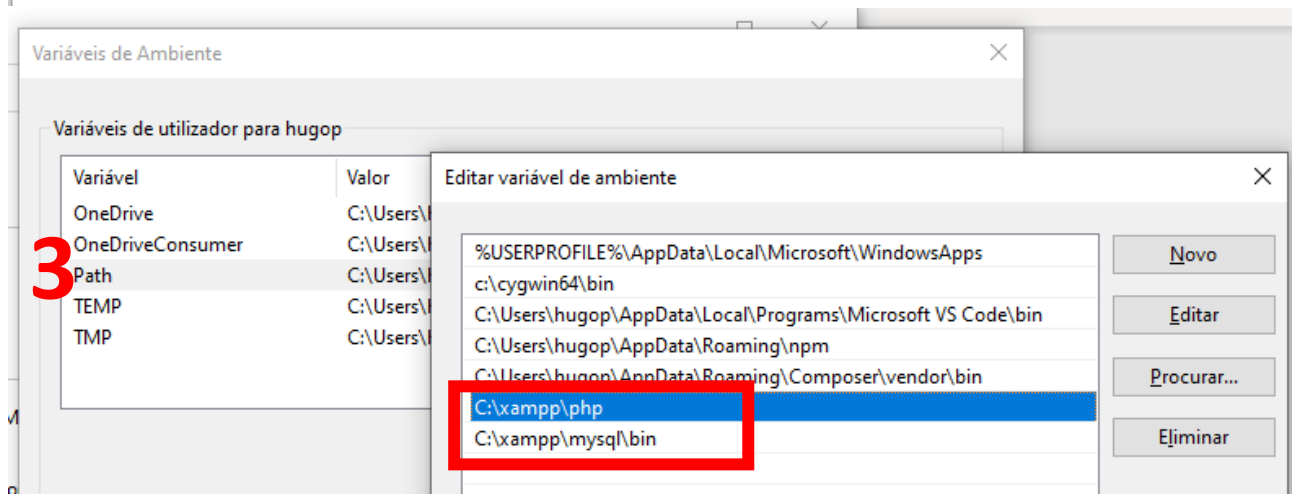
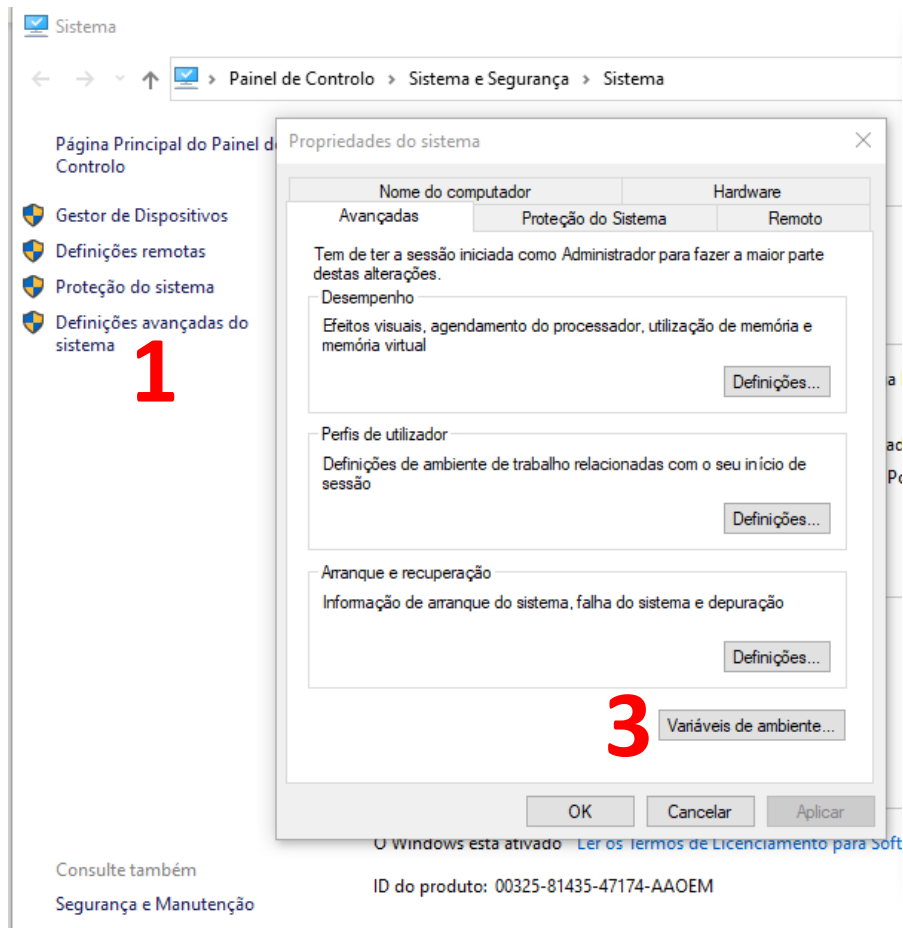
Guião de Criação do Projeto

1) Configuração do Ambiente de Trabalho:

NOTA: Se já tem o ambiente de desenvolvimento configurado passe para o **ponto 2**.

- Instalação do Xampp (<https://www.apachefriends.org>)
- Configuração das Variáveis de Ambiente

Ir a **Computador** → **Propriedades** → **Definições Avançadas (1)** → **Variáveis de Ambiente (2)** → **Path (3)** e verificar se existem as Pastas do PHP e MySQL, se não existirem adicionar:



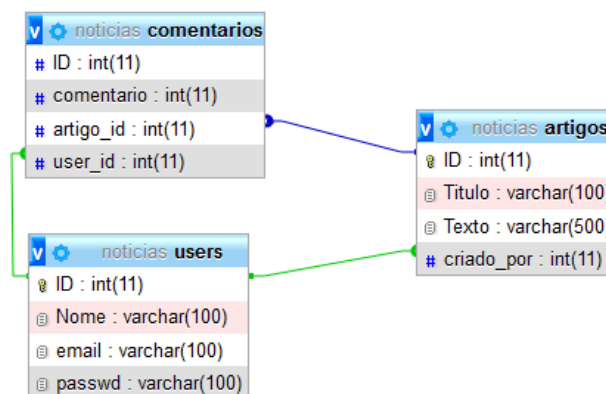
- c. Instalação do composer (<https://getcomposer.org/download/>)
- d. Instalação do Node.js (<https://nodejs.org/>)
- e. Verificação, na linha de comandos executar:
 - i. php -v
 - ii. composer -V
 - iii. node -v
 - iv. npm -v
- f. Instalação do laravel
 - i. Executar na linha de comandos:
composer global require "laravel/installer"
NOTA: também pode ser usado para atualizar a versão do installer do laravel.

2) Criação do Projeto

- a. Executar no CMD:
 - i. `Laravel new nome_do_projeto`

3) Desenho das classes

- a. Pensar nas classes a utilizar e desenhar um diagrama ER com as classes a utilizar



- b. Criar as Migrations (**ver Exemplo Parte 2**) com as chaves estrangeiras (**ver Exemplo Parte 4**).
- c. Antes de executar as migrations criar a Base de dados vazia no phpMyAdmin (**ver Exemplo Parte 2**).
- d. Criar os Modelos (**ver Exemplo Parte 2**) com chaves estrangeiras (**ver Exemplo Parte 4**)

4) Definir Ações (Pode ser em papel)

- a. Definir quais as ações que vão ser criadas:
Ex: posso querer só que um objeto seja apenas visível ou depois de criado não possa ser alterado, isso vai afetar as ações disponíveis.
NOTA: As decisões devem ser justificadas no relatório
 A tabela a seguir mostra as ações mais comuns e as Routes, caminhos e Metodos Http normalmente associados:

Http	URL	Action	Route
GET	/artigos	artigosController.index()	artigos.index
GET	/artigos/create	artigosController.create()	artigos.create
POST	/artigos	artigosController.store()	artigos.store
GET	/artigos/{id}	artigosController.show()	artigos.show
GET	/artigos/{id}/edit	artigosController.edit()	artigos.edit
PUT/PATCH	/artigos/{id}	artigosController.update()	artigos.update
DELETE	/artigos/{id}	artigosController.destroy()	artigos.destroy

NOTA: Os nomes apresentados são convenções, são recomendados, mas não obrigatórios.

- Criar um controller para cada classe, p.ex: artigosController, comentariosController (**ver Exemplo Parte 1**);
- Em cada controller criar os métodos associados a cada ação.
Ex: método index() deve ir buscar todos os artigos e retornar a view correspondente passando os artigos obtidos;
- Criar as views associadas (**ver Exemplo Parte 1**) a cada ação.
- Criar as Routes (**ver Exemplo Parte 1** / **ver Exemplo Parte 3** para Named Routes) associadas que executam os métodos correspondentes.

5) Testar funcionalidades