CRUD com PDO MySQL

fatec – senai centro | Belo Horizonte

CRUD: PDO, Mysql e XAMPP

RUBEM CÂNDIDO

2021

Sumário

[Introdução 2](#_Toc62125182)

[Ferramentas necessárias 2](#_Toc62125183)

[XAMPP 2](#_Toc62125184)

[Editor de texto. 2](#_Toc62125185)

[Start o XAMPP 2](#_Toc62125186)

[Criando uma instancia no banco de dados 3](#_Toc62125187)

[Criando a classe 5](#_Toc62125188)

[Criando uma função para conectar com o banco 10](#_Toc62125189)

[try catch 10](#_Toc62125190)

[PDO 11](#_Toc62125191)

[Index.php 12](#_Toc62125192)

[include() 12](#_Toc62125193)

[Teste Unitário - Conexão 14](#_Toc62125194)

[var\_dump() 14](#_Toc62125195)

[Simulando erros: 15](#_Toc62125196)

[Git 20](#_Toc62125197)

[Conclusão 20](#_Toc62125198)

# Introdução

O propósito desse tutorial é criar um CRUD utilizando PDO e MySQL.

CRUD é um acrónimo do inglês Create, Read, Update e Delete. Representam as quatro operações básicas Inserir, Buscar, Atualizar e Deletar dados, utilizadas em banco de dados relacionais.

PDO acrónimo do inglês para *PHP Data Object* é uma extensão orientada a objetos da linguagem PHP para acesso a banco de dados.

XAMPP é um software que automatiza a instalação de programas necessários para gerar um ambiente de desenvolvimento.

MySQL ou MariaDB é um sistema de gerenciamento de banco de dados.

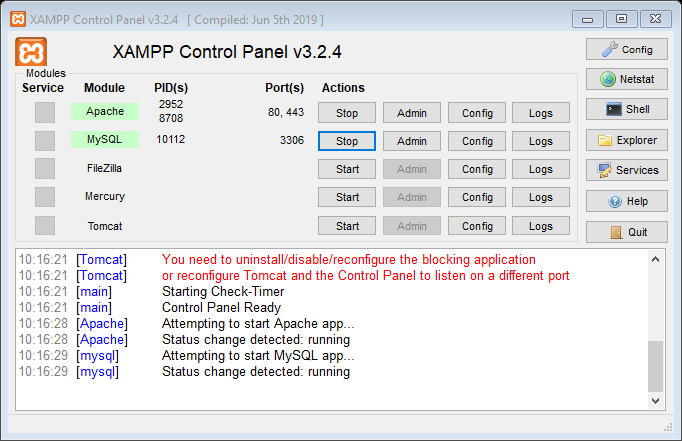
# Ferramentas necessárias

## XAMPP

## Editor de texto.

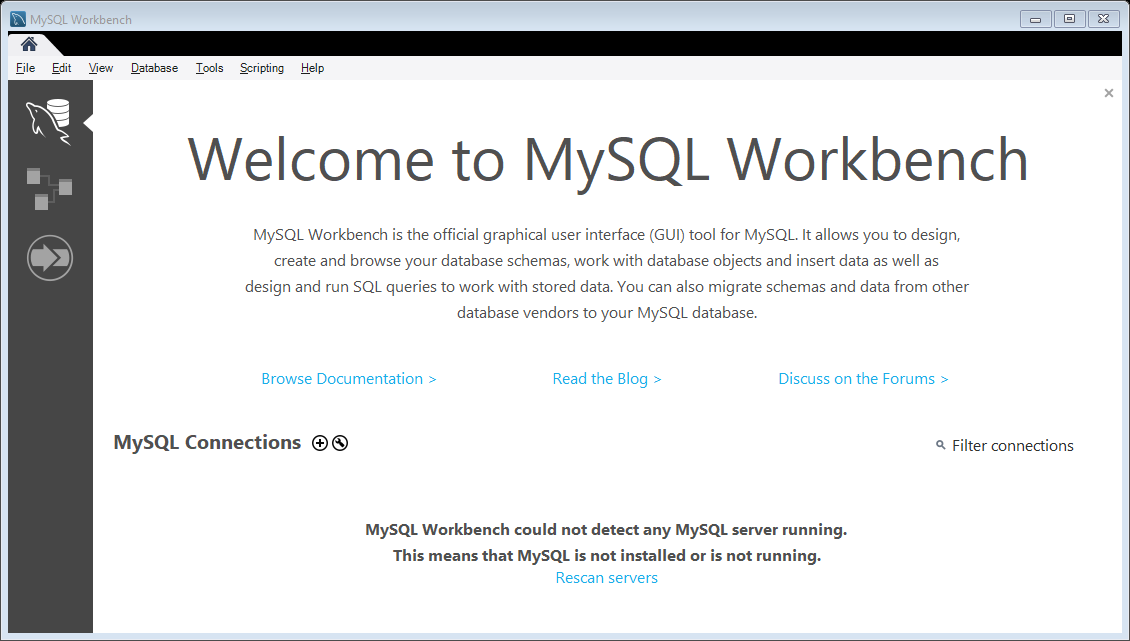
# Start o XAMPP

Para criar um servidor web, precisamos de iniciar o Apache. Para termos acesso à um SGBD, inicie o MySQL.

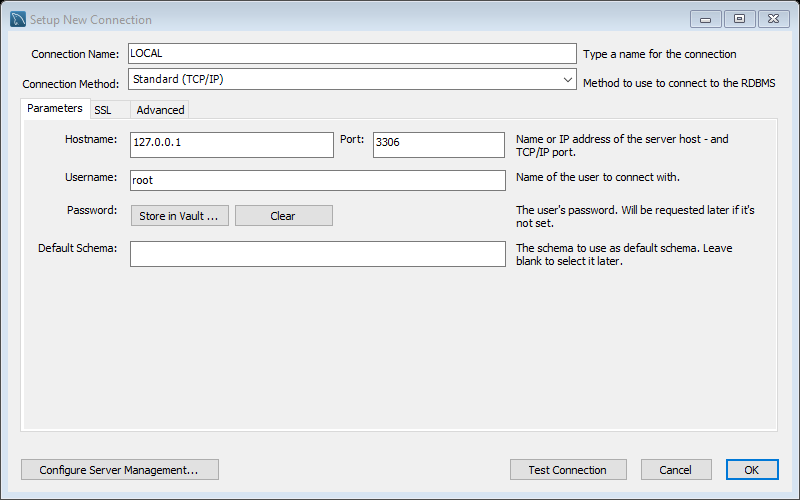


# Criando uma instancia no banco de dados

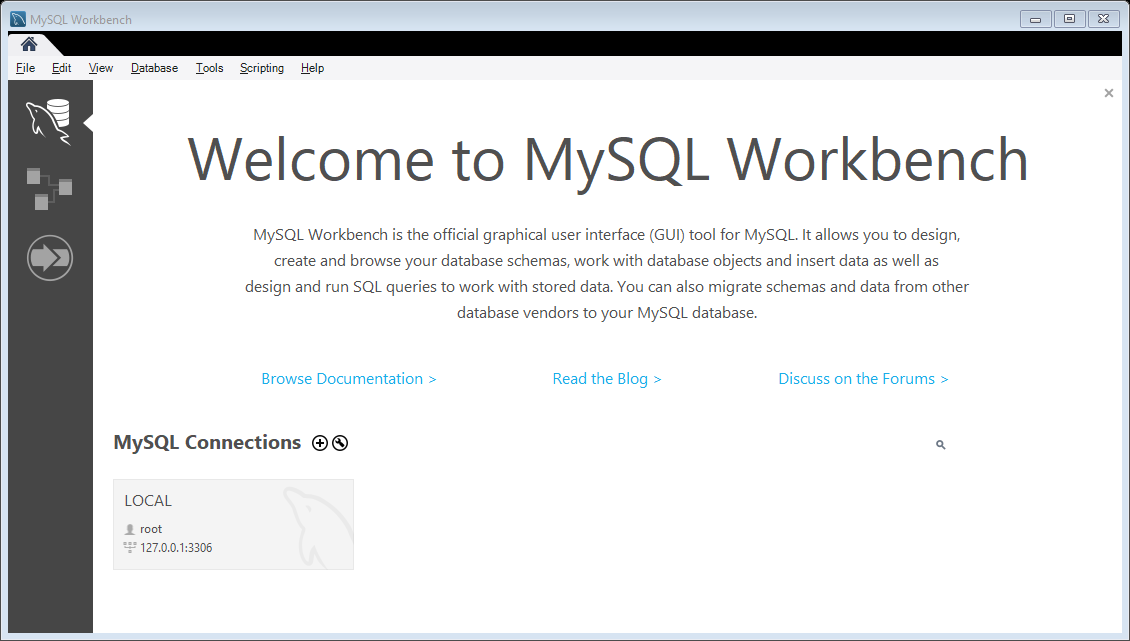
Abra o Workbench e crie uma conexão clicando em MySQL Connection +



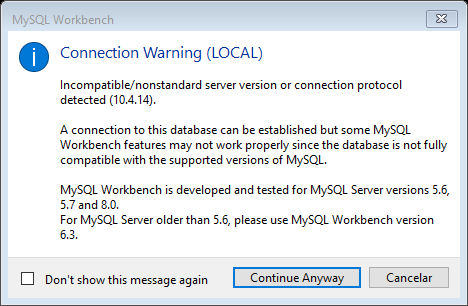
Para uma conexão local e padrão, basta nome.



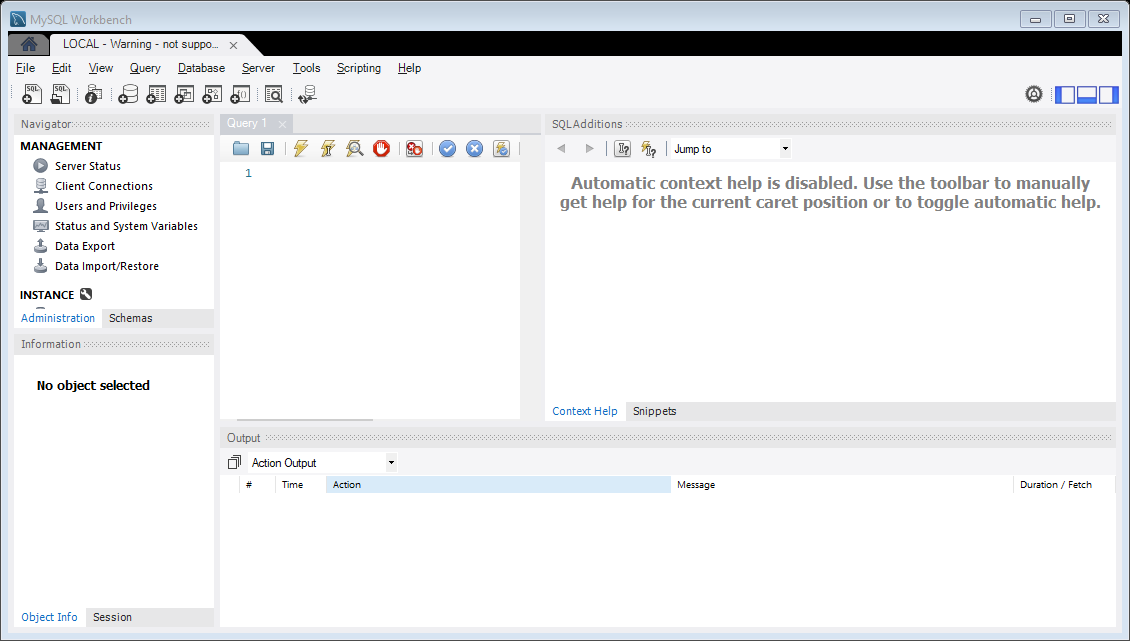
Ao clicar em OK. Duplo clique na conexão



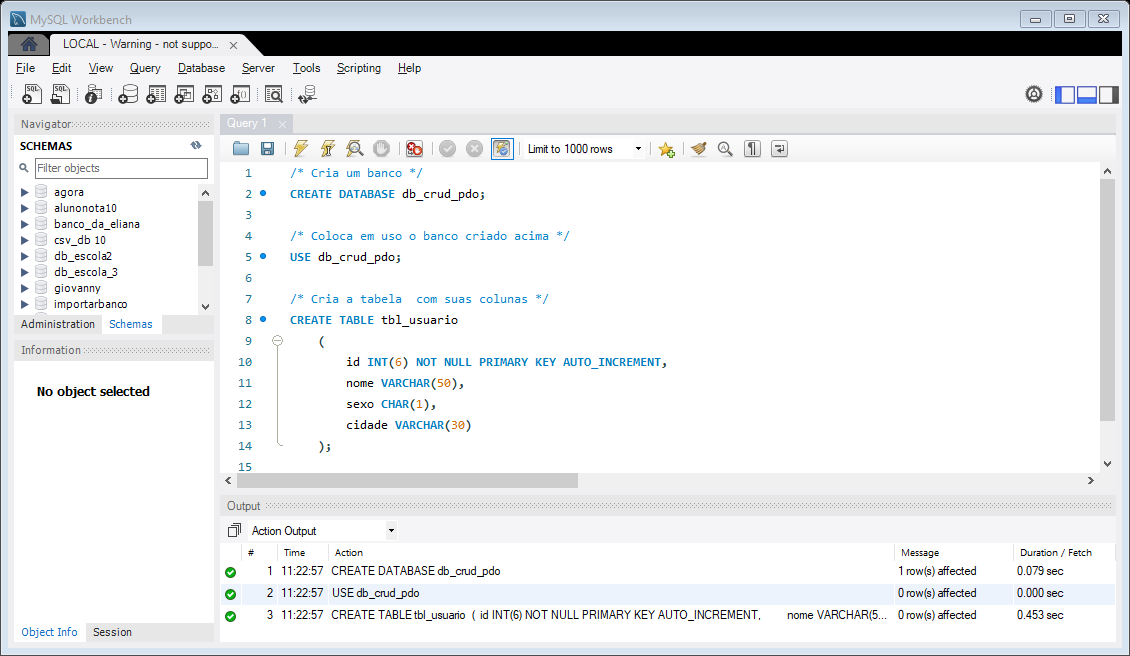
A primeira vez que o workbench se conecta no MySQL. Marque Continue Anyway.



Interface do Workbench

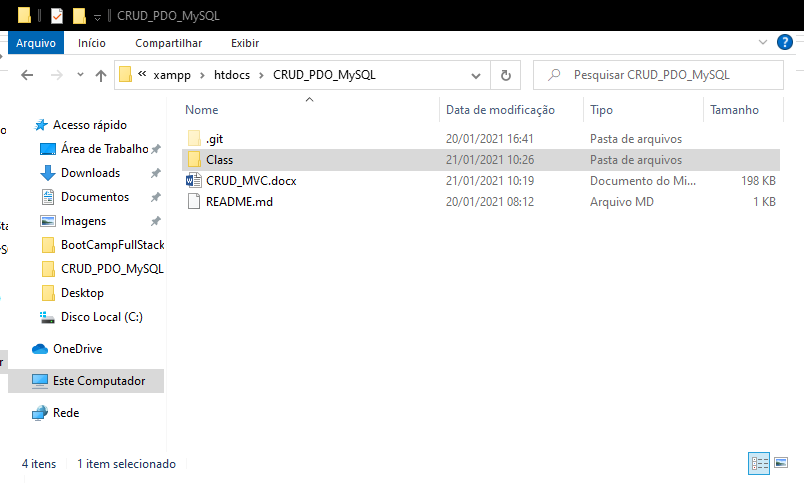


Criando o banco e uma tabela com suas colunas

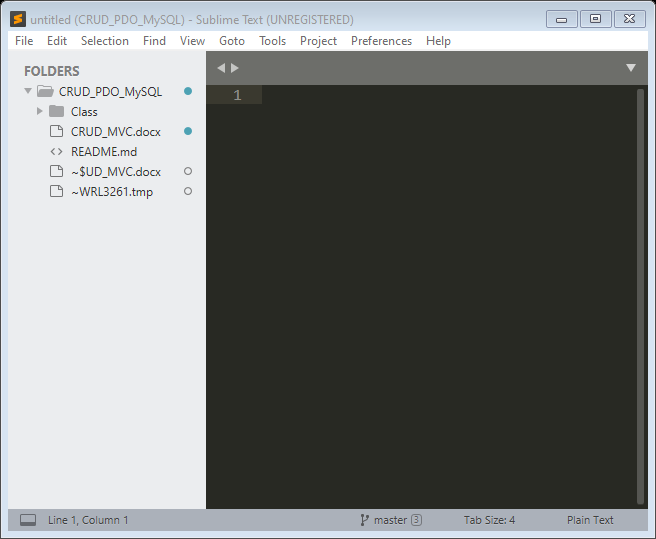


# Criando a classe

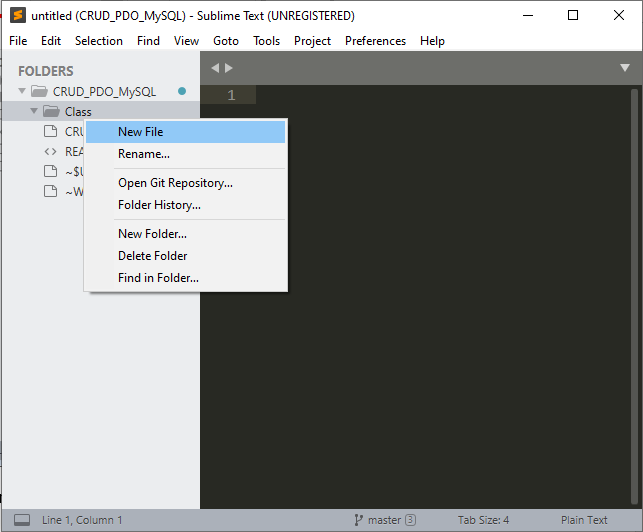
Crie um diretório chamado Class na raiz do seu projeto.



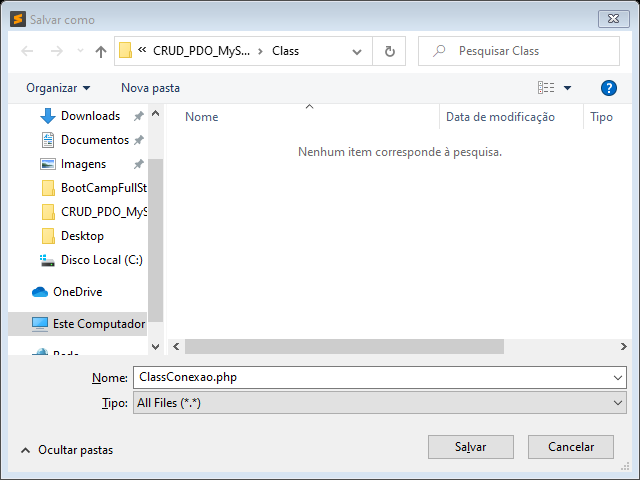
Abra o folder com seu editor de texto.



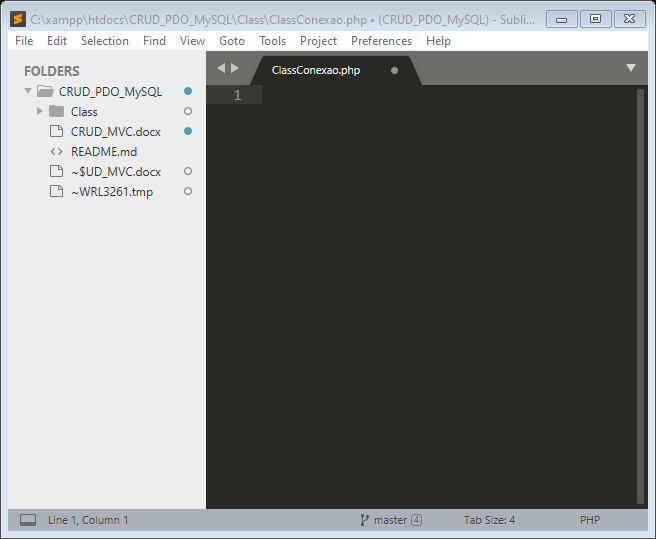
Crie um arquivo chamado ClassConexao.php

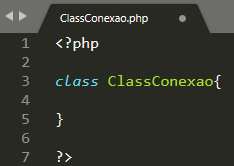


Salvando com o atalho Ctrl + S

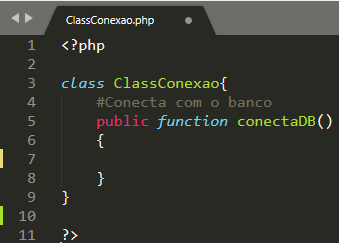


Arquivo criado





# Criando uma função para conectar com o banco



## try catch

Vamos tratar possíveis erros com um try catch.

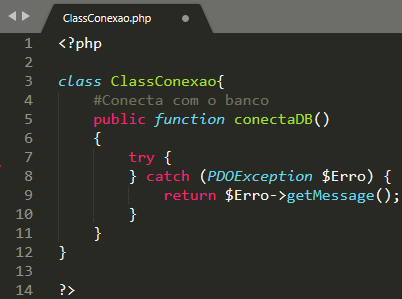
Atalho: Digite try e pressione a tecla tab para auto completar



Vamos nomear de acordo com nossa aplicação

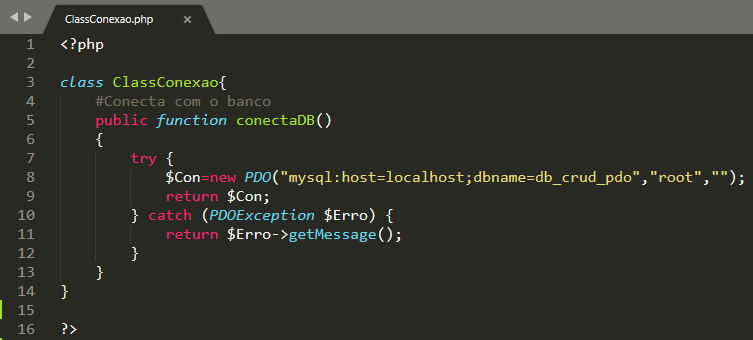


O catch irá retornar uma mensagem de erro caso a conexão falhe.



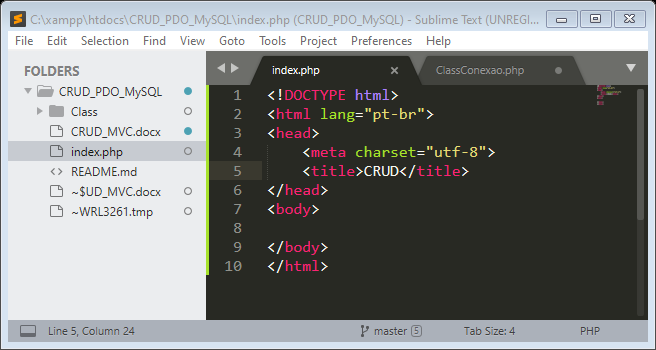
## PDO

Criando a variável de conexão com PDO



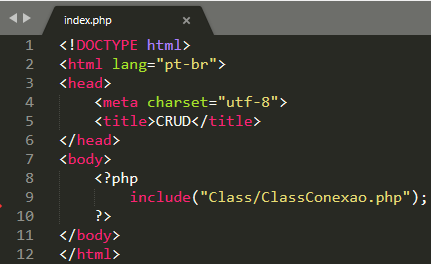
# Index.php

Na raiz do projeto, crie um arquivo chamado index.php com o esqueleto padrão do HTML5

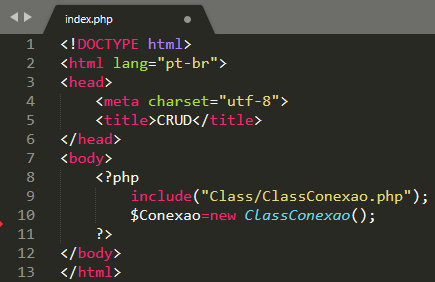


# include()

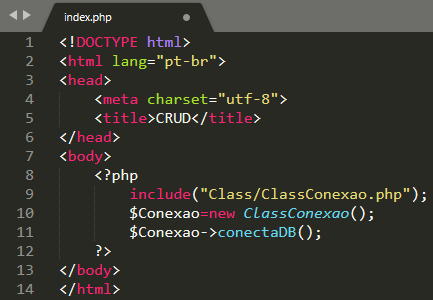
Com a função include, vamos referenciar a classe para dentro do index.php



Instanciar a classe ClassConexao()

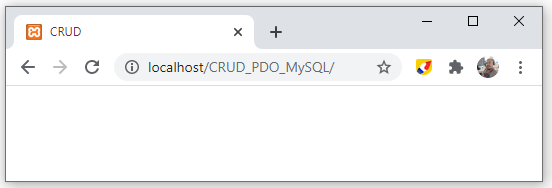


Vamos chamar a função



# Teste Unitário - Conexão

Testando a conexão, vamos abrir o navegador e digitar o caminho do nosso projeto: http://localhost/CRUD\_PDO\_MySQL/

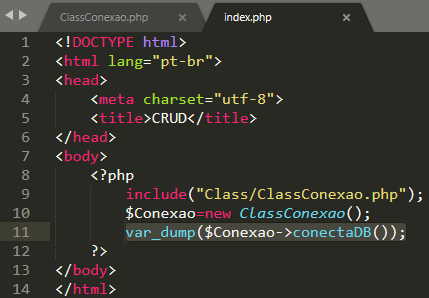


Tudo certo.

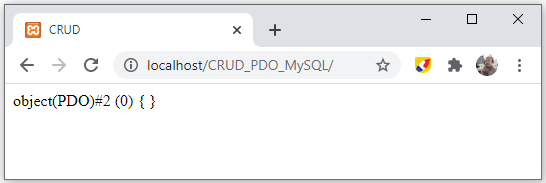
Vamos simular alguns erros:

## var\_dump()

Acrescente a função var\_dump() para debugar a conexão.

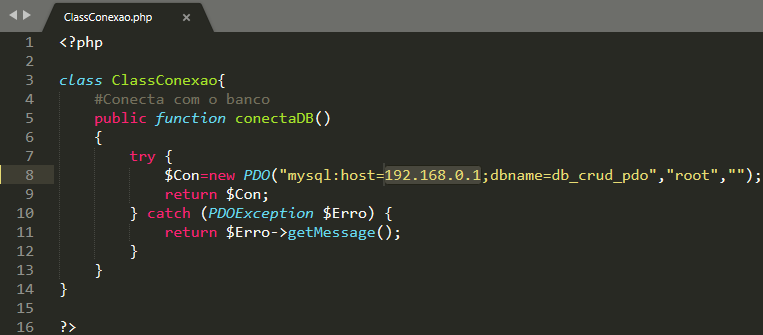


Atualize a página para ver o resultado esperado:

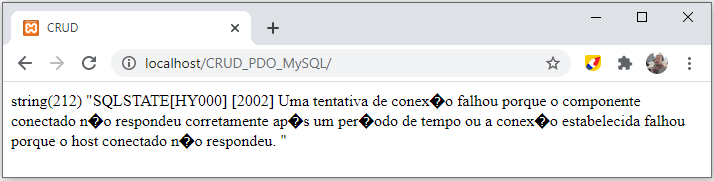


## Simulando erros:

Altere o host

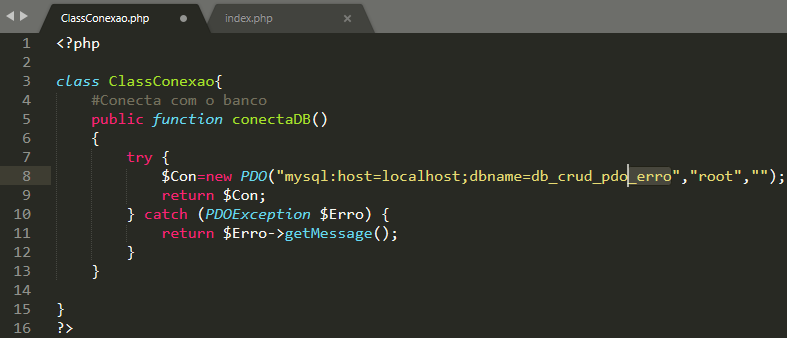


Atualize a página:

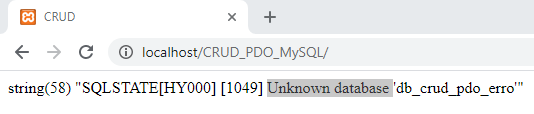


Volte o host para localhost.

Altere o nome do banco para forçar um erro

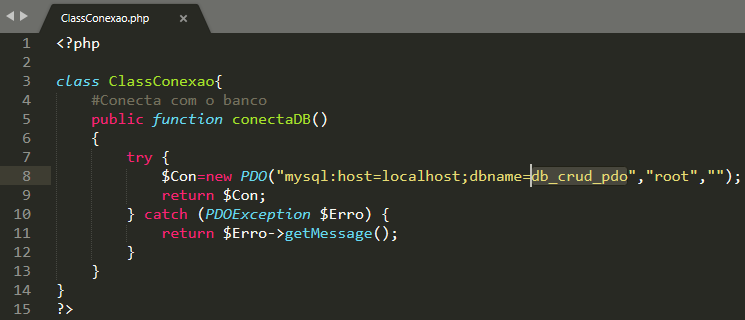


Atualize a página

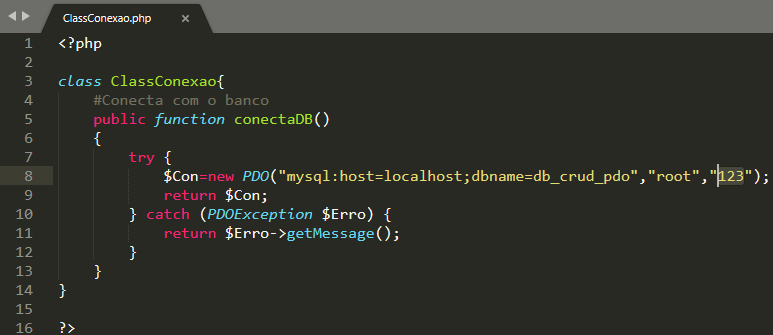


Banco de dados não encontrado.

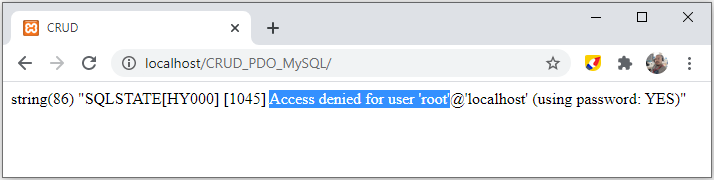
Corrija o nome do banco



Agora insira uma senha incorreta

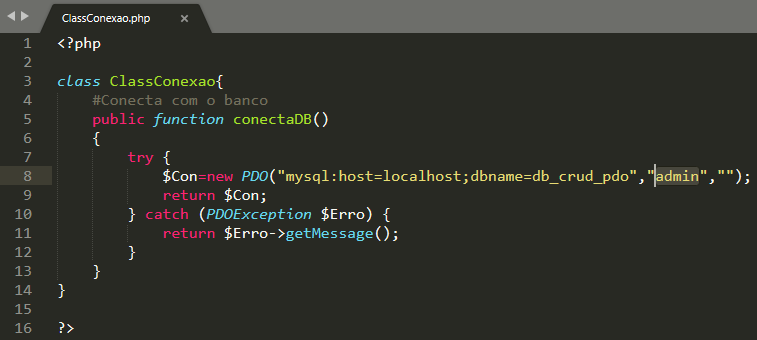


Atualize a página:

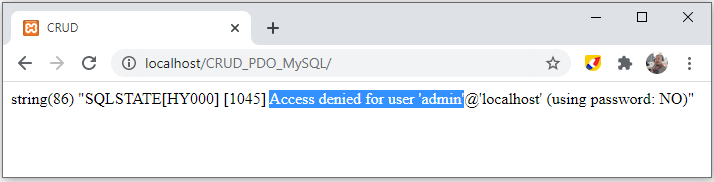


Acesso negado para o usuário root

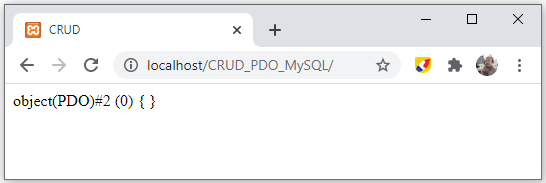
Corrija a senha e altere o usuário:



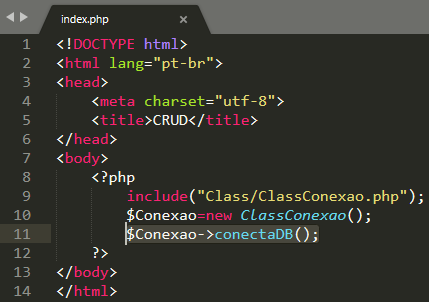
Atualize a página:



Mensagem esperada



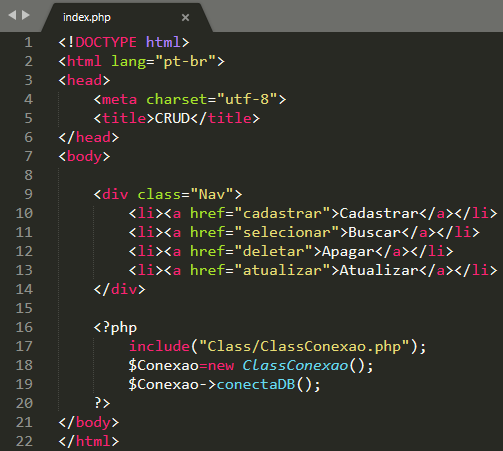
Retire o var\_dump



# Layout

Menu de navegação

No arquivo index.php, acima do include, crie uma div com uma classe Nav. Dentre da classe em formato de lista, insira os links.

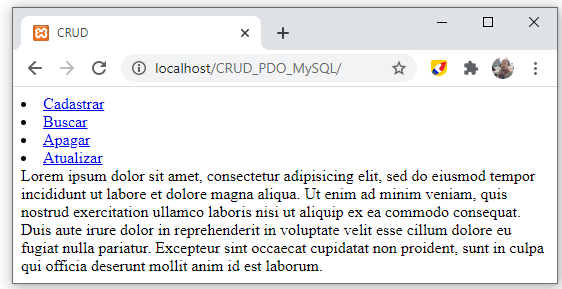


Classe Content – Para o conteúdo

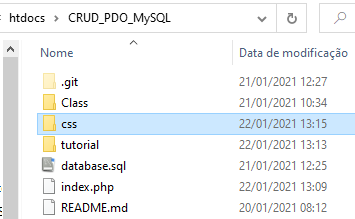


Atalho: O texto foi gerado digitando logem e pressionando a tecla Tab do teclado

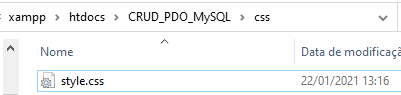
Resultado:



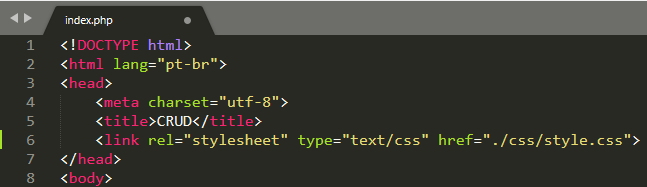
## CSS

Crie um diretório na raiz do seu projeto chamdo css. 

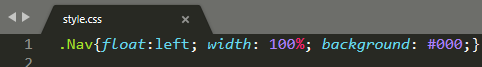
Dentro da pasta css, crie um arquivo chamado style.css



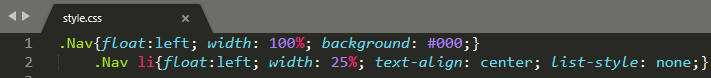
Faça o arquivo index.php reconhecer o style.css



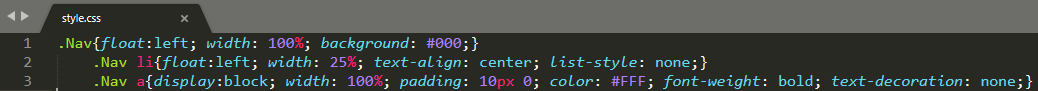
Abra o arquibo style.css no editor de texto e coloque um estilo na classe Nav



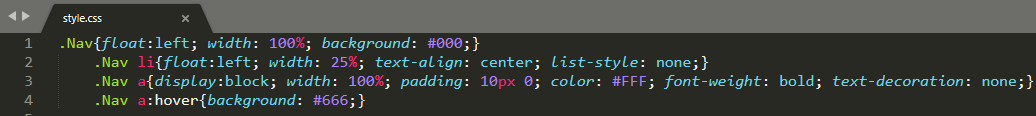
Elemento li



Elemento a



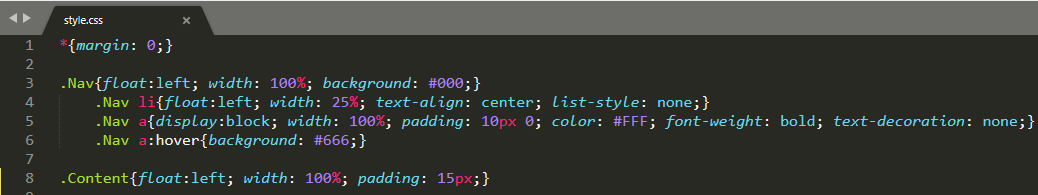
a:hover



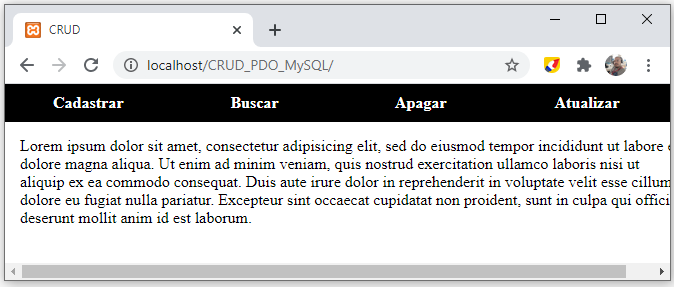
Reset margin



.Content

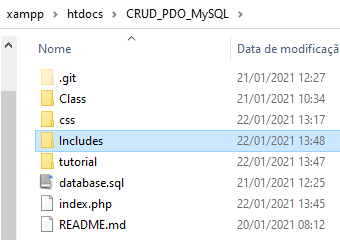


Resultado:

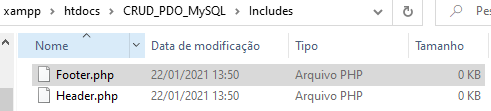


# Organizando o projeto

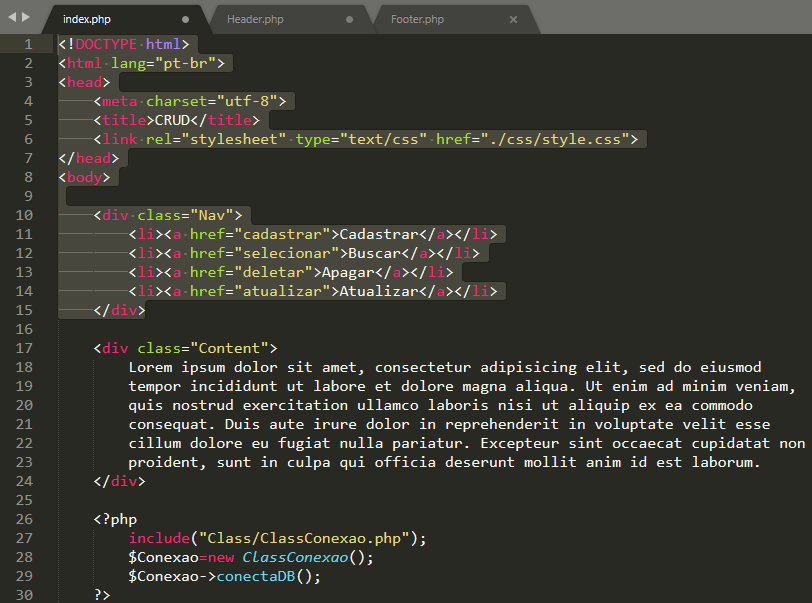
No diretório do projeto, crie um diretório chamado includes



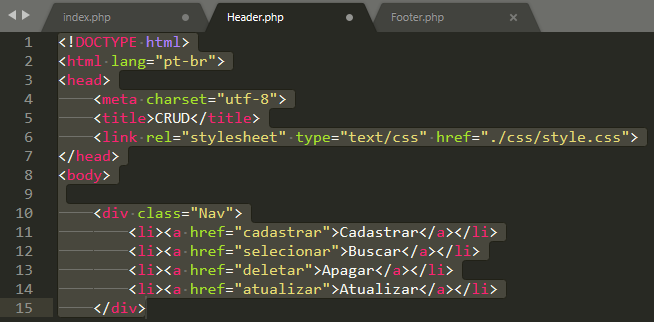
Dentro do diretório Includes crie os arquivos Header.php e o Footer.php



Do index.php, recorte a parte superior da página

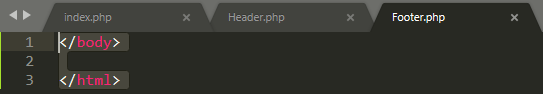


E coloque no arquivo Header.php



Pegue o final da página “rodapé” do arquivo index.php 

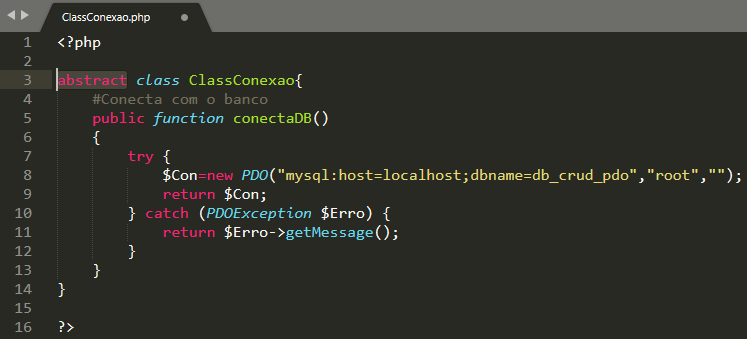
e coloque no Footer.php



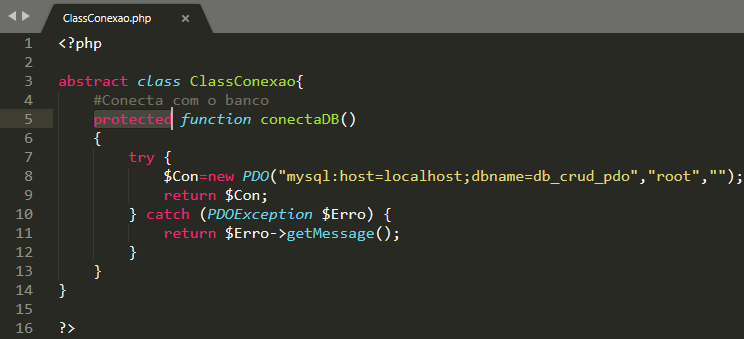
No index.php vamos deixar apenas o que se repete em todas as páginas. Ou seja, os links.

# Abstract

Por questões de segurança, vamos transformar a ClassConexao em uma classe abstrata, para que ela não possa ser instanciada, apenas estendida.



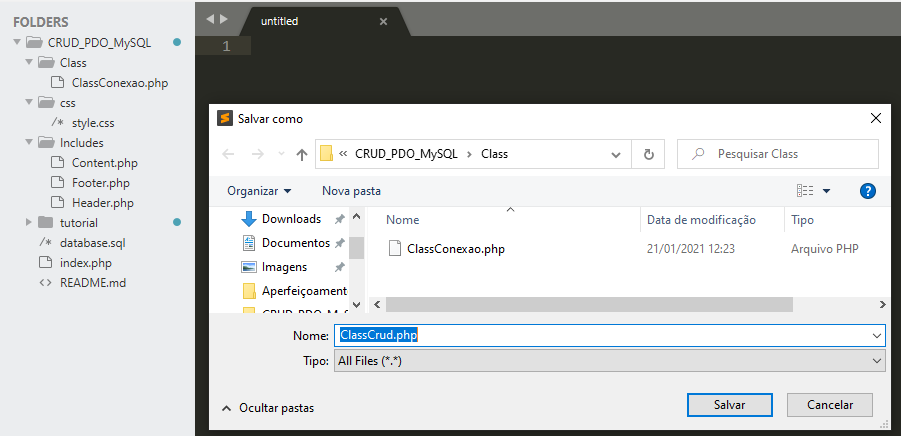
Dentro da classe ClassConexao, vamos alterar o modificador de acesso da função conectaDB de public para **protected**.



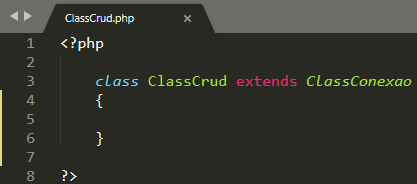
# Prepared Statements

Evita SQL Inject, implementando uma camada a mais de segurança ao se tratar do Banco de Dados.

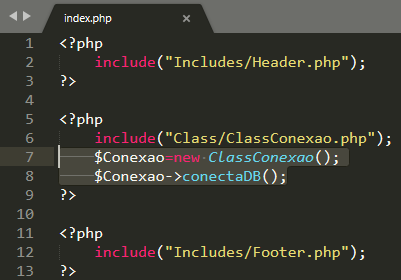
No diretório Class, crie um arquivo chamado ClassCrud.php



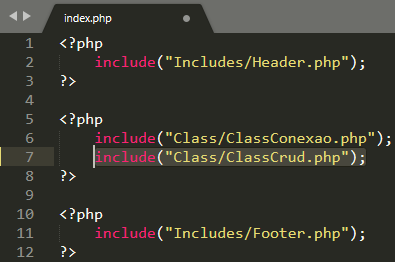
Vamos estender a ClassConexao



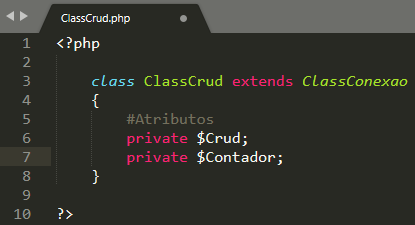
No arquivo index.php. Apague as variáveis



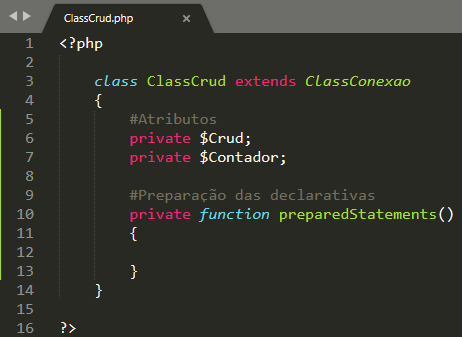
Acrescente a ClassCrud com um include



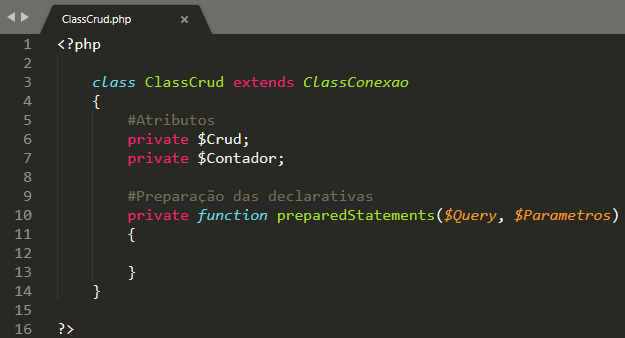
Crie dois atributos na ClassCrud. $Crud e $Contador



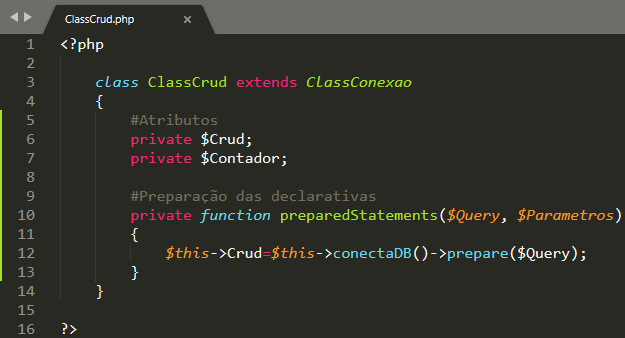
Crie um método para a preparação das declarativas



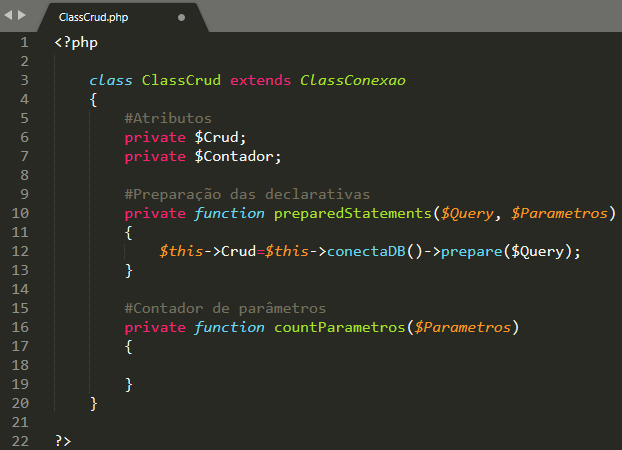
Esse método irá receber duas variáveis como argumentos



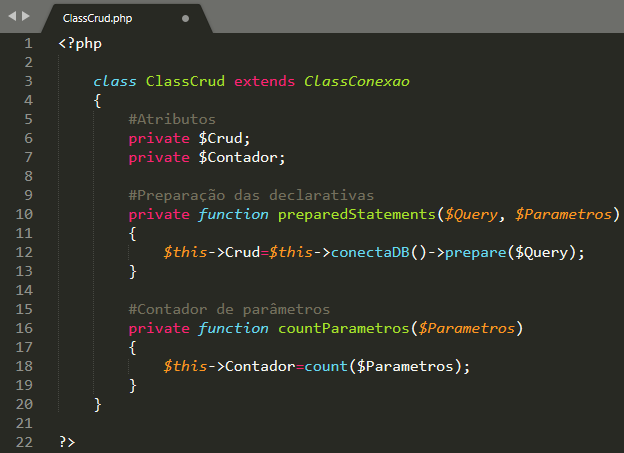
O atributo Crud irá receber o método de conexão conectaDB prepara uma Query.



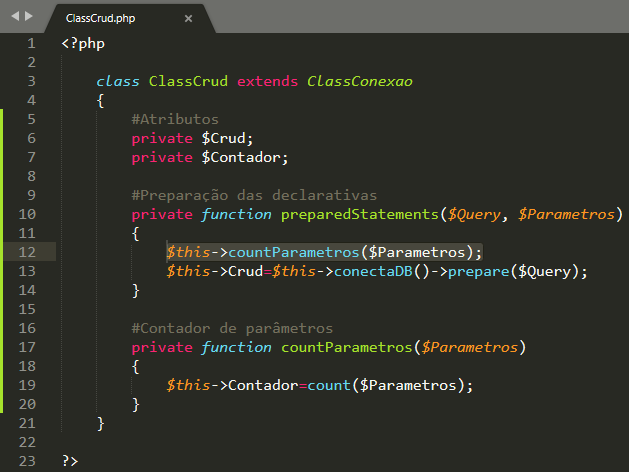
Crie um contador de parâmetros, para os bindValue().



Essa função irá atribuir ao contador uma contagem dos parâmetros.

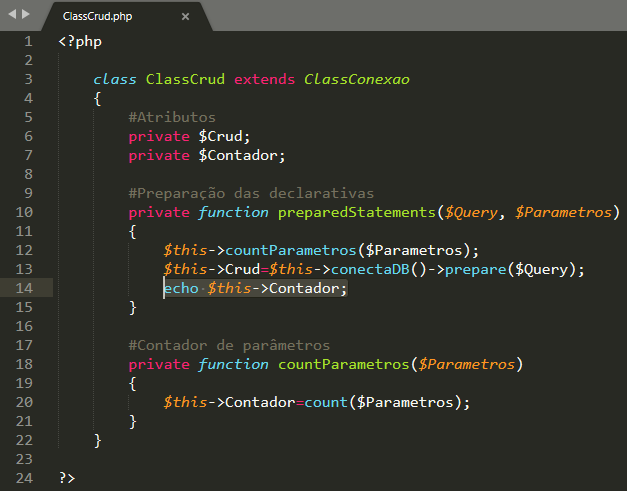


No preparedStatements vamos chamar essa função passando o $Parametros como argumento.

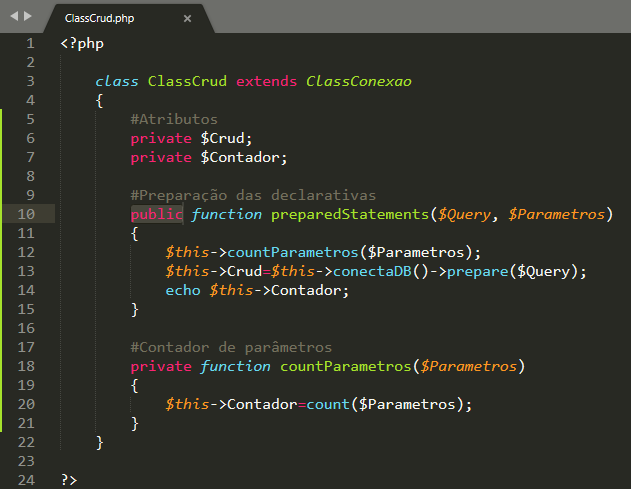


## Teste unitário

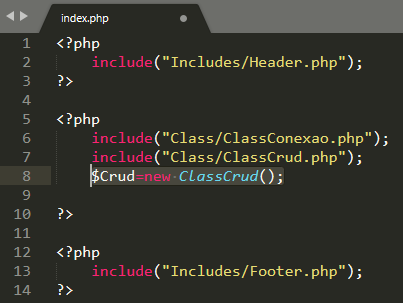
Vamos dar um echo para ver o que está acontecendo.



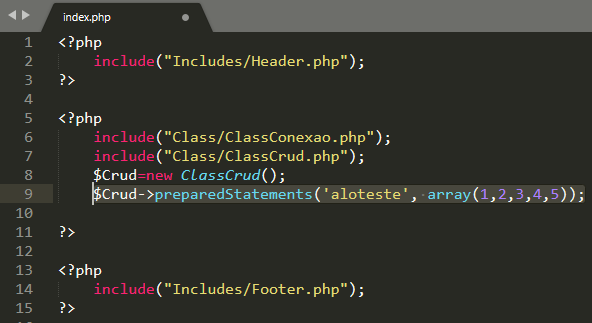
Altere o modificador de acesso para public



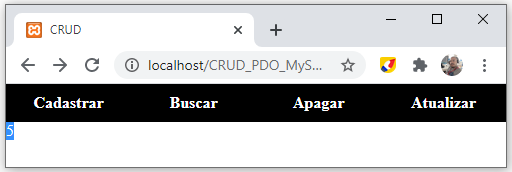
No index.php, instancie



Passe alguns argumentos



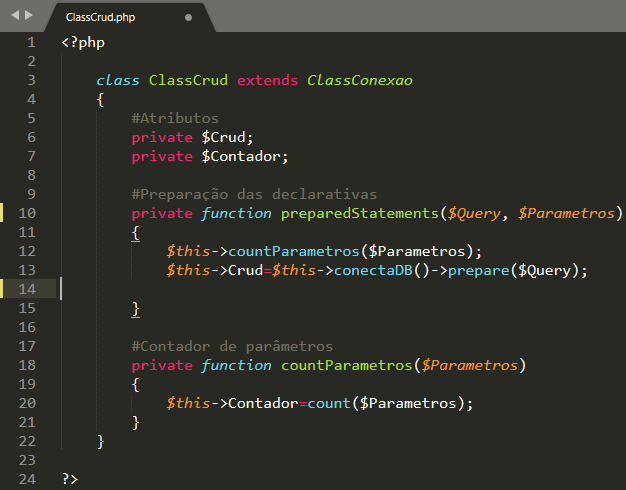
Verificando:



Voltando ao normal. Apague a linha 9 do index.php

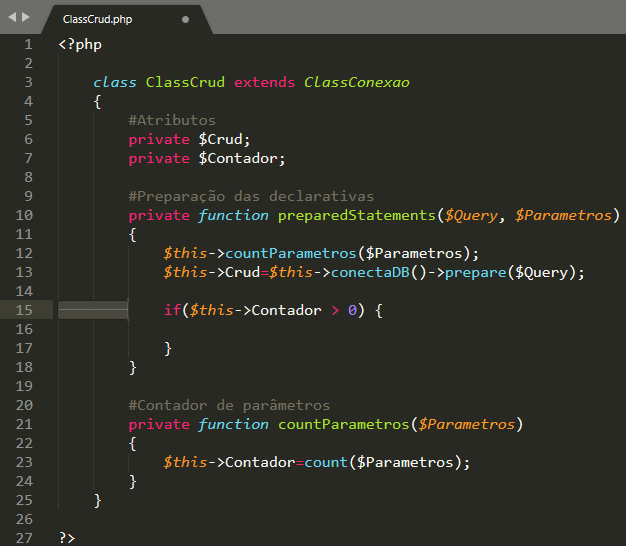


Volte o modificador de acesso para private no ClassCrud e apague o echo na linha 14.

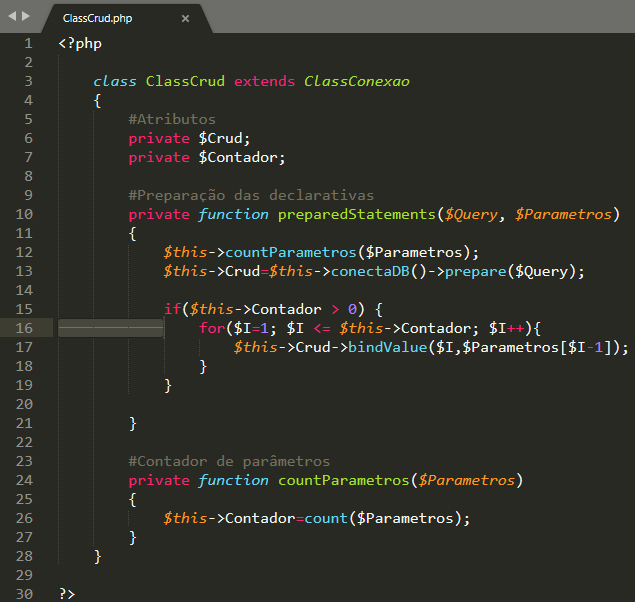


# bindValue()

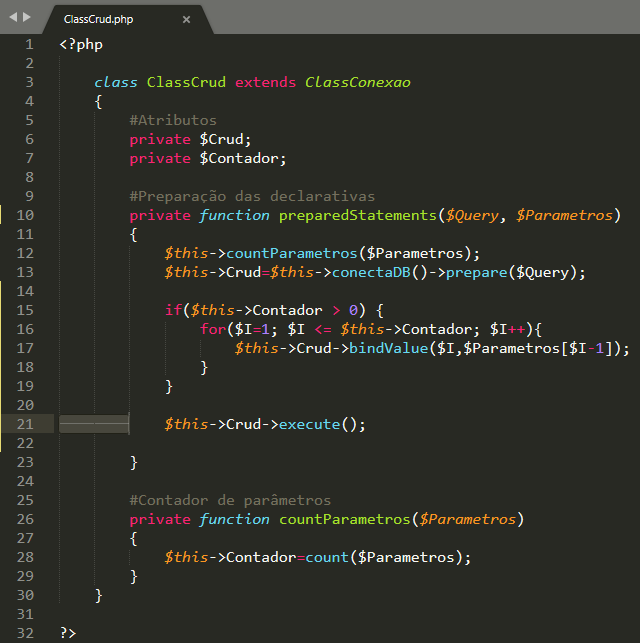
No arquivo ClassCrud.php. Crie uma condição if, para o Contador ter efeito apenas quando for maior que 0.



Crie um for para realizar o bildValue



Caso a query não retorne nada a função pula para o execute()



# Git

Endereço do repositório no git:

https://github.com/endroni/CRUD\_PDO\_MySQL.git

# Conclusão

Uma estruturação em quatro camadas pode ser um tanto complexo no início, porém possibilita um grande reaproveitamento em trabalhos que possuem funções semelhantes.

Espero que tenha valia para seus estudos.

**BIBLIOGRAFIA BÁSICA:**

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7. ed. São Paulo: Pearson Makron Books, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison Wesley, 2007.

TONSIG, Sergio Luiz. **Engenharia de Software: análise e projeto de sistemas**. 2. ed. Rio de Janeiro: Ciência Moderna, 2008.

**BIBLIOGRAFIA COMPLEMENTAR:**

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Rio de Janeiro: Elsevier, 2007.

UDEMY. **Programando em MVC com C#.** Hugo Vasconcelos