

# Curso Laravel 5.4 (Aulas 61 a 67)

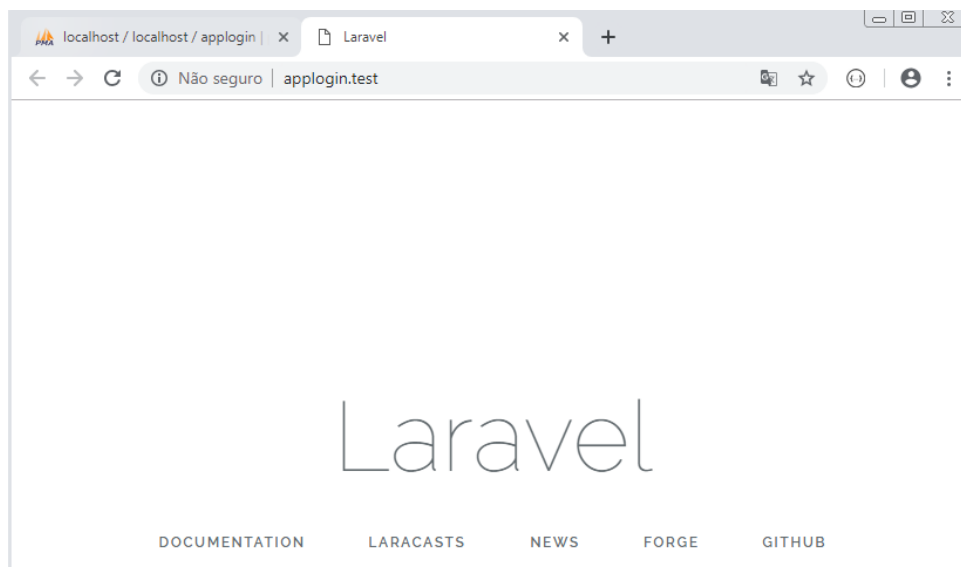
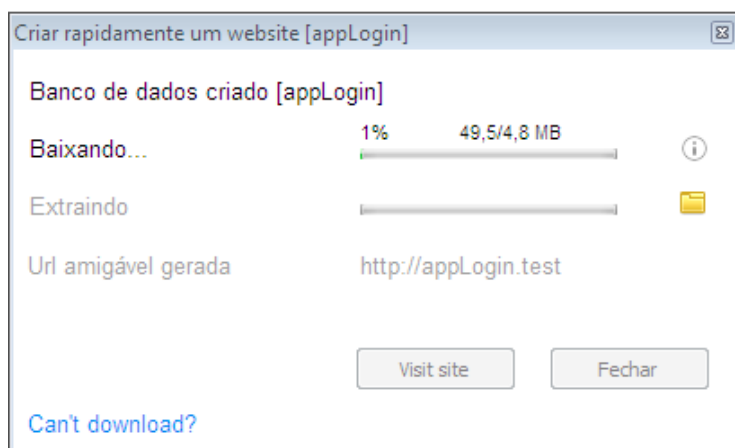
## SYS4SOFT (João Ribeiro)

[https://www.youtube.com/watch?v=0T5gM1WRNsY&list=PLXik\\_5Br-zO893qVjjP7a4gg4NYrl33w1&index=1](https://www.youtube.com/watch?v=0T5gM1WRNsY&list=PLXik_5Br-zO893qVjjP7a4gg4NYrl33w1&index=1)

Resumo do curso feito por Roberto Pinheiro

## Aula 061 – Projeto para exercício de login - Parte 1

No Laragon, clicar com o botão direito na área de trabalho e selecionar "**Criar rapidamente um website**" → "**Laravel (zip)**". Digitar como nome do projeto: appLogin e clicar no botão Ok.

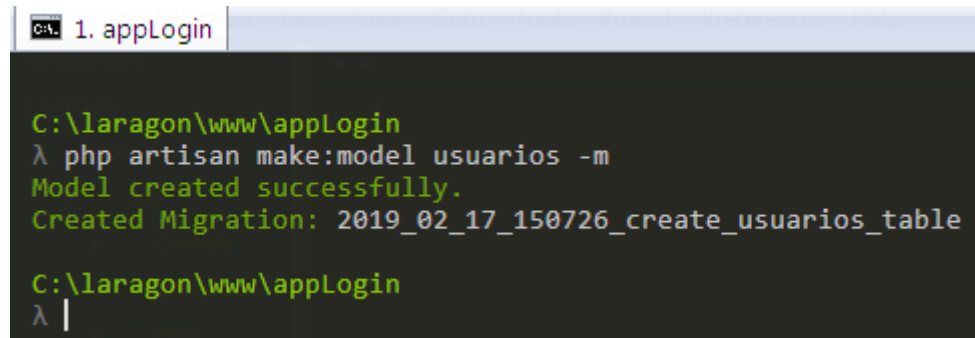


No Sublime Text, apague as duas migrations padrões e na pasta **app** apague o arquivo **User.php**.

## Criando o model e a migration

No terminal crie o model e a migration, entrando com o seguinte comando:

```
php artisan make:model usuarios -m
```



```
C:\laragon\www\appLogin
λ php artisan make:model usuarios -m
Model created successfully.
Created Migration: 2019_02_17_150726_create_usuarios_table

C:\laragon\www\appLogin
λ |
```

**Note:** Before getting started, make sure that your `users` (or equivalent) table contains a nullable, string `remember_token` column of 100 characters. This column will be used to store a token for "remember me" sessions being maintained by your application. This can be done by using `$table->rememberToken();` in a migration.

database/migrations/2019\_02\_17\_150726\_create\_usuarios\_table.php

```
<?php
```

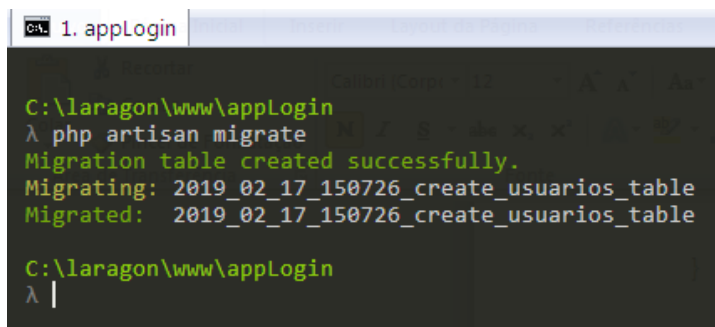
```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateUsuariosTable extends Migration
{
    public function up()
    {
        Schema::create('usuarios', function (Blueprint $table) {
            $table->increments('id_usuario');
            $table->string('usuario');
            $table->string('senha');
            $table->string('email');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('usuarios');
    }
}
```

No terminal, crie a tabela, com o seguinte comando:

```
php artisan migrate
```



```
C:\laragon\www\appLogin
λ php artisan migrate
Migration table created successfully.
Migrating: 2019_02_17_150726_create_usuarios_table
Migrated: 2019_02_17_150726_create_usuarios_table

C:\laragon\www\appLogin
λ |
```

Server: localhost:3306 » Database: applogin » Table: usuarios

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Triggers](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id_usuario	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 2	usuario	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 3	senha	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 4	email	varchar(255)	utf8mb4_unicode_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 5	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 6	created_at	timestamp			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 7	updated_at	timestamp			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

☐ Check all
 With selected:
 [Browse](#)
[Change](#)
[Drop](#)
[Primary](#)
[Unique](#)
[Index](#)
[Fulltext](#)

## Criando o controller

No terminal, entre com o seguinte comando:

```
php artisan make:controller usuariosController
```

```

1. appLogin
C:\laragon\www\appLogin
λ php artisan make:controller usuariosController
Controller created successfully.

C:\laragon\www\appLogin
λ

```

**app/http/Controllers/usuariosController.php**

```
<?php
```

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
```

```
class usuariosController extends Controller
{
```

```

    // Apresentar o formulário de Login
    public function ApresentarFormularioLogin(){

        return view('usuarios_login');
    }
}

```

## Criando as views

### resources/views/layouts/app.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>appLogin</title>

    <link href="{{asset('css/app.css')}}" rel="stylesheet">

    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>

    <div class="container">
      @yield('conteudo')
    </div>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <script src="{{asset('js/app.js')}}"></script>
  </body>
</html>
```

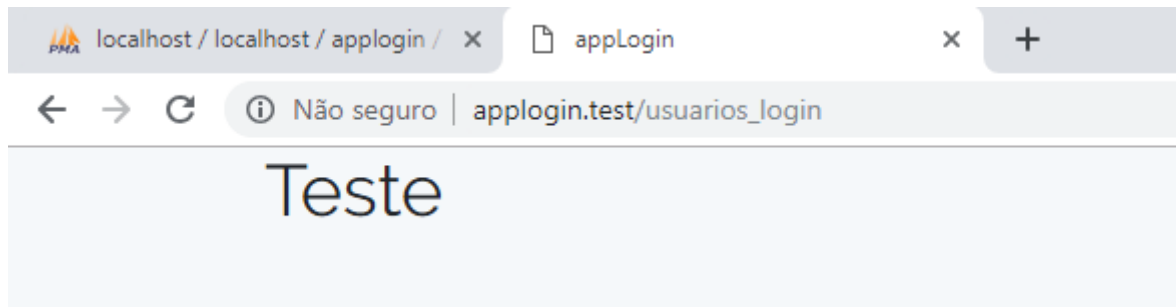
### resources/views/usuarios\_login.blade.php

```
@extends('layouts.app')

@section('conteudo')
  <h1>Teste</h1>
@endsection
```

## Criando a route

```
Route::get('/usuarios_login', 'usuariosController@ApresentarFormularioLogin');
```



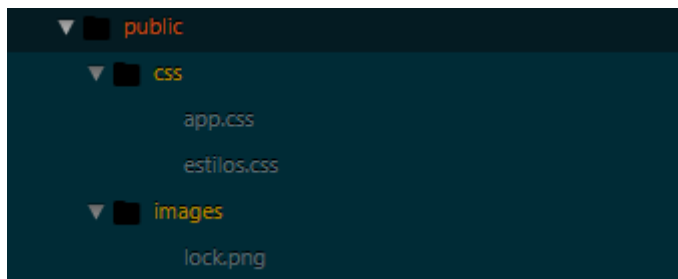
## Aula 062 – Projeto para exercício de login - Parte 2

**public/css/estilos.css**

```
#img-lock{  
    margin: 50px auto;  
}
```

**resources/views/layouts/app.blade.php**

```
<!DOCTYPE html>  
<html lang="pt-br">  
  <head>  
    <meta charset="utf-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <title>appLogin</title>  
    <link href="{{asset('css/app.css')}}" rel="stylesheet">  
    <link href="{{asset('css/estilos.css')}}" rel="stylesheet">  
    <!--[if lt IE 9]>  
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>  
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>  
    <![endif]-->  
  </head>  
  <body>  
  
    <div class="container">  
      @yield('conteudo')  
    </div>  
  
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>  
    <script src="{{asset('js/app.js')}}"></script>  
  </body>  
</html>
```



**resources/views/usuarios\_login.php**

@extends('layouts.app')

@section('conteudo')

```
<div class="row">
    
</div>

<div class="row">

    <div class="col-md-4">

    </div>

    <div class="col-md-4">
        <form>
            <div class="form-group">
                <label for="usuario">Usuário:</label>
                <input type="text" class="form-control" name="text_usuario" id="usuario"
placeholder="Usuário" required>
            </div>
            <div class="form-group">
                <label for="senha">Senha:</label>
                <input type="password" class="form-control" name="text_senha" id="senha"
placeholder="Senha" required>
            </div>
            <button type="submit" class="btn btn-primary">Login</button>
        </form>
    </div>

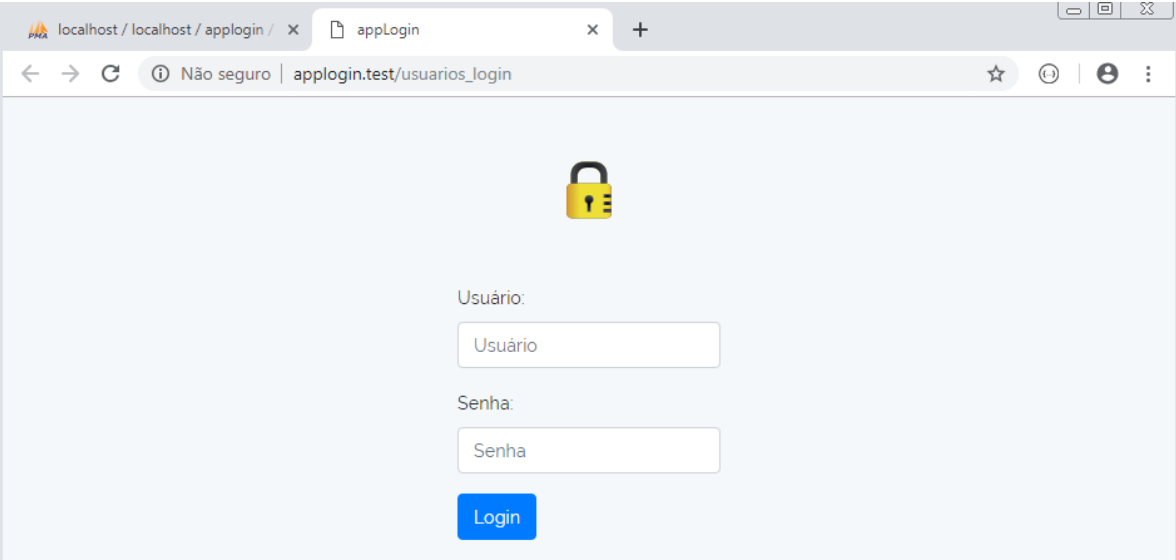
    <div class="col-md-4">

    </div>

</div>

@endsection
```





## Aula 063 – Validação de formulários - Parte 1

**routes/web.php**

```
Route::get('/usuarios_login', 'usuariosController@ApresentarFormularioLogin');  
Route::post('/fazer_login', 'usuariosController@FazerLogin');
```

## resources/views/usuarios\_login.blade.php

```
@extends('layouts.app')
```

```
@section('conteudo')
```

```
<div class="row">
    
</div>
```

```
<div class="row">
    <div class="col-md-4"></div>
```

```
<div class="col-md-4">
```

```
    {{-- Erros de validação --}}
    @if (count($errors) != 0)
        @foreach ($errors->all() as $erro)
            <p class="alert alert-danger">{{$erro}}</p>
        @endforeach
```

```
    {{-- {{count($errors)}} --}}
```

```
@endif
```

```
<form method="POST" action="/fazer_login">
```

```
    {{-- csrf --}}
    {{csrf_field()}}
```

```
    <div class="form-group">
        <label for="usuario">Usuário:</label>
        <input type="text" class="form-control" name="text_usuario" id="usuario"
placeholder="Usuário">
    </div>
    <div class="form-group">
        <label for="senha">Senha:</label>
        <input type="password" class="form-control" name="text_senha" id="senha"
placeholder="Senha">
    </div>
    <button type="submit" class="btn btn-primary">Login</button>
</form>
```

```
</div>
```

```
<div class="col-md-4"></div>
</div>
```

```
@endsection
```

**app/http/Controllers/usuariosController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class usuariosController extends Controller
{

    // Apresentar o formulário de Login
    public function ApresentarFormularioLogin(){
        return view('usuarios_login');
    }

    public function FazerLogin(Request $request){

        // Verificação dos inputs

        $this->validate($request,[
            'text_usuario' => 'required',
            'text_senha' => 'required|min:3'

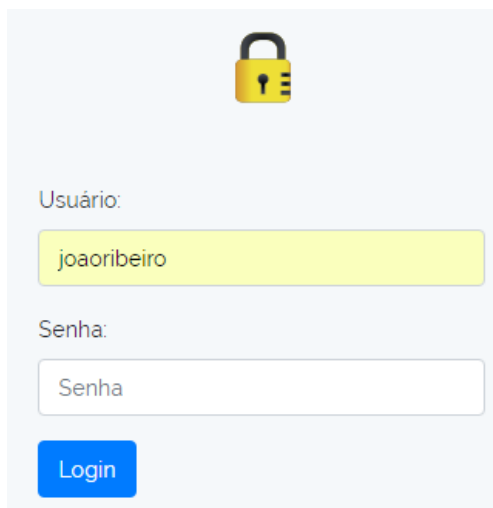
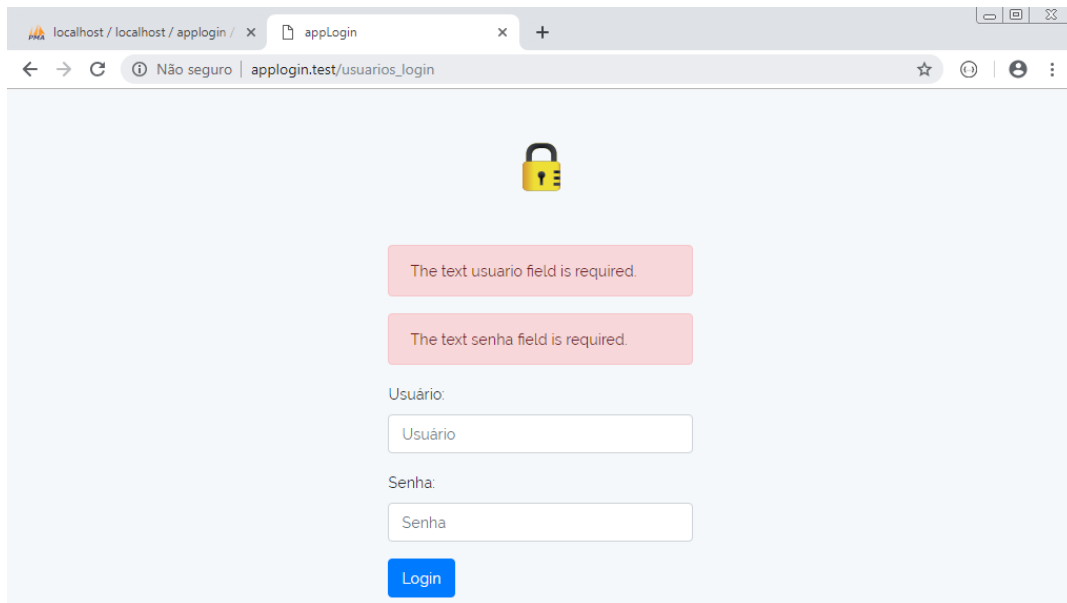
        ]);

        // Só depois da verificação, consultar a BD

        return 'Validado';

    }

}
```





Usuário:

joaoribeiro

Senha:

..

Login



The text senha must be at least 3 characters.

Usuário:

Usuário

Senha:

Senha

Login

## Aula 064 – Validação de formulários - Parte 2

resources/views/usuarios\_login.blade.php

```
@extends('layouts.app')
```

```
@section('conteudo')
```

```
<div class="row">
    
</div>
```

```
<div class="row">
```

```
    <div class="col-md-4"></div>
```

```
    <div class="col-md-4">
```

```
        {{-- Erros de validação --}
        @include('inc.erros')
```

```
        <form method="POST" action="/fazer_login">
```

```
            {{-- csrf --}}
            {{csrf_field()}}
```

```
            <div class="form-group">
                <label for="usuario">Usuário:</label>
                <input type="text" class="form-control" name="text_usuario" id="usuario"
placeholder="Usuário">
            </div>
            <div class="form-group">
                <label for="senha">Senha:</label>
                <input type="password" class="form-control" name="text_senha" id="senha"
placeholder="Senha">
            </div>
            <button type="submit" class="btn btn-primary">Login</button>
        </form>
```

```
    </div>
```

```
    <div class="col-md-4"></div>
```

```
</div>
```

```
@endsection
```

### **resources/views/inc/erros.blade.php**

```
{{-- Controle de erros do formulário --}}
@if (count($errors) != 0)
    @foreach ($errors->all() as $erro)
        <p class="alert alert-danger">{{$erro}}</p>
    @endforeach
@endif
```

### **app/http/Controllers/usuariosController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class usuariosController extends Controller
{

    // Apresentar o formulário de Login
    public function ApresentarFormularioLogin(){
        return view('usuarios_login');
    }

    public function FazerLogin(Request $request){

        // Verificação dos inputs

        $this->validate($request,[
            'text_usuario' => 'alpha|required|min:5',
            'text_senha' => 'required|min:3'

        ]);

        // Só depois da verificação, consultar a BD

        return 'Validado';

    }

}
```





Usuário:

Senha:

Login



The text usuario may only contain letters.

The text senha field is required.

Usuário:

Senha:

Login

## Aula 065 – Criação de senhas encriptadas - Hashing

app/http/Controllers/usuariosController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Hash;
```

```
class usuariosController extends Controller  
{
```

```
    // Apresentar o formulário de Login  
    public function ApresentarFormularioLogin(){  
        return view('usuarios_login');  
    }
```

```
    public function FazerLogin(Request $request){
```

```
        $senha = $request->text_senha;
```

```
        $senha_encriptada = Hash::make($senha);
```

```
        return "Senha: " . $senha . "<br>Senha encriptada: " . $senha_encriptada;
```

```
    }
```

```
}
```



Usuário:

Senha:

Login



localhost / localhost / applogin / x

applogin.test/fazer\_login x



Não seguro | applogin.test/fazer\_login

Senha: abcdef

Senha encriptada: \$2y\$10\$36ws8r/Pbp1TANZ619kc6u8CIUk9gBUk6dErRf6tmZkfvQ...xezEm

## Aula 066 – Inserir usuários na base de dados

**app/http/Controllers/usuariosController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use App\usuarios;

class usuariosController extends Controller
{
    // Apresentar o formulário de Login
    public function ApresentarFormularioLogin(){
        return view('usuarios_login');
    }

    public function FazerLogin(Request $request){
    }

    public function InserirUsuario(){

        // Inserir um usuário na base de dados

        // $novo = new usuarios;
        // $novo->usuario = 'João';
        // $novo->senha = Hash::make('abc123');
        // $novo->email = 'teste@teste.com';
        // $novo->save();

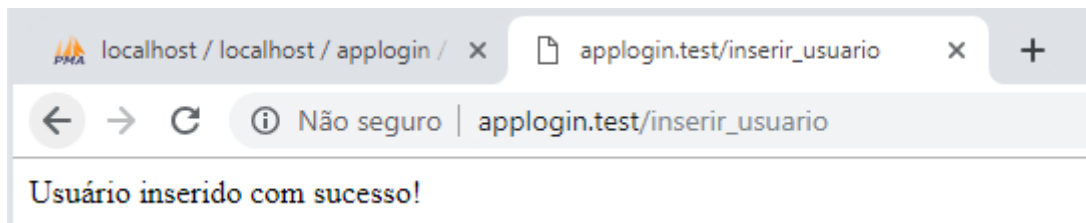
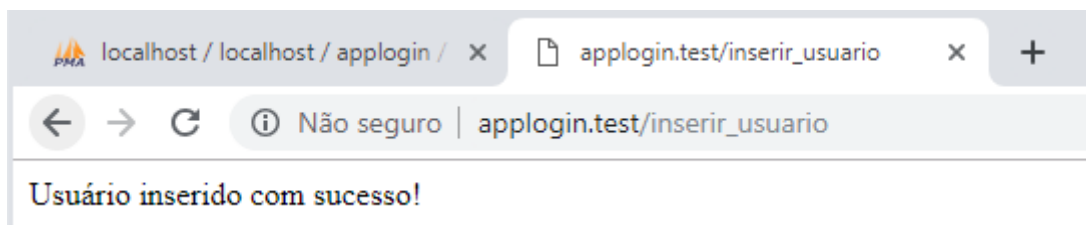
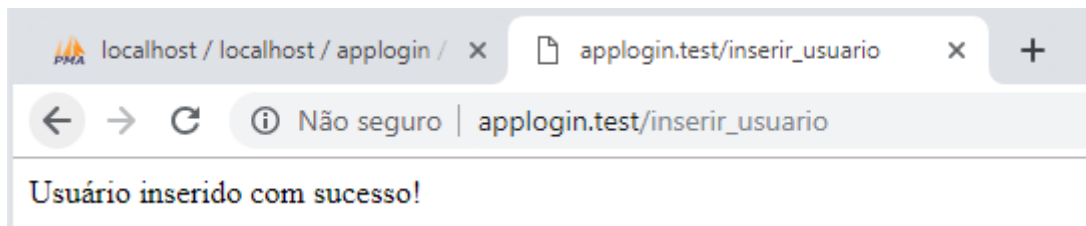
        // $novo = new usuarios;
        // $novo->usuario = 'Carlos';
        // $novo->senha = Hash::make('minhapass');
        // $novo->email = 'carlos@teste.com';
        // $novo->save();

        $novo = new usuarios;
        $novo->usuario = 'Ana';
        $novo->senha = Hash::make('adivinhaesta');
        $novo->email = 'ana@teste.com';
        $novo->save();

        return 'Usuário inserido com sucesso!';
    }
}
```

## routes/web.php

```
Route::get('/usuarios_login', 'usuariosController@ApresentarFormularioLogin');
Route::post('/fazer_login', 'usuariosController@FazerLogin');
// temp
Route::get('/inserir_usuario', 'usuariosController@InserirUsuario');
```



Showing rows 0 - 2 (3 total, Query took 0.0010 seconds.)

SELECT \* FROM `usuarios`

Profiling [Edit inline] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

		id_usuario	usuario	senha	email	remember_token	created_at
<input type="checkbox"/>	Edit Copy Delete	1	João	\$2y\$10\$py/6tvSV8Jyp2nhnYCPyOei4Oj0qheQ3lkcGOEu2ki...	teste@teste.com	NULL	2019-02-17 22:36:31
<input type="checkbox"/>	Edit Copy Delete	2	Carlos	\$2y\$10\$VUi2ecYpNWiHCgZVvS.UkOK3rDvF/RIC3dg9w9d.0S4...	carlos@teste.com	NULL	2019-02-17 22:43:33
<input type="checkbox"/>	Edit Copy Delete	3	Ana	\$2y\$10\$gmHA3GbLPc2HqgHNJFU8d.ol4iWe4O97qZLc2slhcIN...	ana@teste.com	NULL	2019-02-17 22:45:41

Check all | With selected: Edit Copy Delete Export

## Aula 067 – Verificar login de usuário

**app/http/Controllers/usuariosController.php**

```
public function FazerLogin(Request $request){
    // Verifica se os dados de login estão corretos

    // Primeiro vai buscar a conta do usuário

    $dados = usuarios::where('usuario', $request->text_usuario)->first();

    if($dados == null){
        $num_ocorrencias = 0;
    } else {
        $num_ocorrencias = $dados->count();
    }

    $resultado = "";

    if ($num_ocorrencias == 0) {
        $resultado = "Não existe essa conta de usuário";
    } else {

        // foi encontrado o usuário
        if (Hash::check($request->text_senha, $dados->senha)) {
            $resultado = "Login com sucesso";
        } else {
            $resultado = "Login não realizado. A senha está errada";
        }

    }

    return $resultado;
}
}
```



Usuário:

Senha:

Login



localhost / localhost / applogin /



applogin.test/fazer\_login



Não seguro | applogin.test/fazer\_login

Login não realizado. A senha está errada



Usuário:

Senha:

Login



localhost / localhost / applogin /



applogin.test/fazer\_login



Não seguro | applogin.test/fazer\_login

Login com sucesso



Usuário:

Senha:

Login



localhost / localhost / applogin /



applogin.test/fazer\_login



Não seguro | applogin.test/fazer\_login

Não existe essa conta de usuário