

## UNIVERSIDAD TECNOLÓGICA DEL VALLE DE TOLUCA

Dirección de Carrera de Tecnologías de la Información y Comunicación

Ingeniería en Desarrollo y Gestión de Software

**Tarea:** Segundo factor de autenticación en Laravel

**Asignatura:** Seguridad en el desarrollo de aplicaciones

**Profesor:** Mario González

**Estudiante:**

Jiménez Venancio Nancy

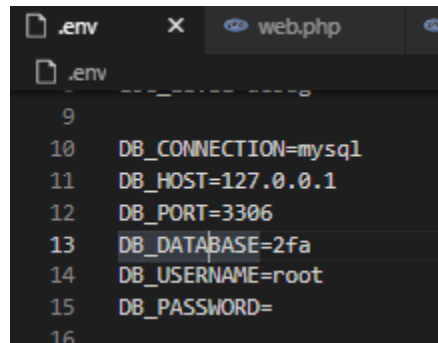
221810614

**Grupo:** IDGS – 83

**Cuatrimestre:** 8°

SANTA MARÍA ATARASQUILLO, LERMA ESTADO DE MÉXICO. MARZO 2021

1. Debemos crear una base de datos, y configurar el archivo **.env**

A screenshot of a code editor showing the .env file. The file contains database configuration settings for MySQL. The settings are: DB\_CONNECTION=mysql, DB\_HOST=127.0.0.1, DB\_PORT=3306, DB\_DATABASE=2fa, DB\_USERNAME=root, and DB\_PASSWORD=. The line DB\_DATABASE=2fa is highlighted with a mouse cursor.

```
.env
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=2fa
14 DB_USERNAME=root
15 DB_PASSWORD=
16
```

2. Nos vamos a la parte de Migraciones ocuparemos el **create\_users\_table** donde agregamos:

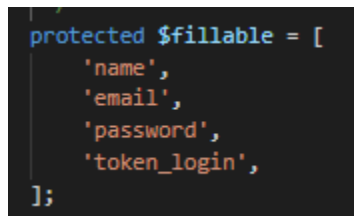
**\$table->string('token\_login')->nullable();**

A screenshot of a code editor showing the create\_users\_table migration file. The file contains a public function up() that creates a 'users' table with columns: id, name, email, email\_verified\_at, password, token\_login, and timestamps. The token\_login column is added with the string() method and nullable() option.

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->string('token_login')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
}
```

3. Y en la parte del Modelo de **User** solo agregamos en la parte:

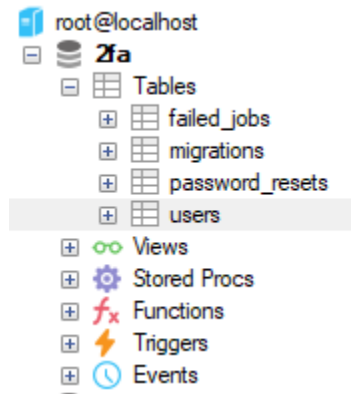
**'token\_login',**

A screenshot of a code editor showing the User model's fillable property. The property is an array containing 'name', 'email', 'password', and 'token\_login'.

```
protected $fillable = [
    'name',
    'email',
    'password',
    'token_login',
];
```

4. Levantamos las migraciones para que las tablas existentes se muestren en la base de datos con:

**php artisan migrate**



5. El comando utilizado para crear el entorno de **ui** y obtener andamios de autenticación en laravel es :

**composer require laravel/ui**

6. Para agregar la configuración básica incluyendo las vistas para registro y autenticación de usuarios, debemos indicar la opción de **--auth** al comando correspondiente:

**php artisan ui vue --auth**

7. Para instalar las dependencias de NPM se ejecuta:

**npm install**

8. Para la compilación de los archivos se ejecuta:

**npm run dev**

9. Instalamos el componente que nos ayudara a que el proyecto tenga el soporte de doble autenticación:

**composer require pragmarx/google2fa-laravel**

10. Y adicionamos el componente de para generar el código QR, instalando:

**composer require bacon/bacon-qr-code**

11. En el archivo de **LoginController.php** ubicado en **app/Http/Controllers/Auth** se agregarán las dependencias que se instalaron para su función.

```
app > Http > Controllers > Auth > LoginController.php
1  <?php
2
3  namespace App\Http\Controllers\Auth;
4
5  use App\Models\User;
6  use App\Providers\RouteServiceProvider;
7  use Illuminate\Http\Request;
8  use PragmaRX\Google2FA\Google2FA;
9  use App\Http\Controllers\Controller;
10 use Illuminate\Support\Facades\Auth;
11 use BaconQrCode\Renderer\ImageRenderer;
12 use BaconQrCode\Writer as BaconQrCodeWriter;
13 use Illuminate\Foundation\Auth\AuthenticatesUsers;
14 use BaconQrCode\Renderer\Image\ImagickImageBackEnd;
15 use BaconQrCode\Renderer\RendererStyle\RendererStyle;
16
```

12. Creando ellos siguientes métodos para la autenticación del formulario:

```
30 use AuthenticatesUsers;
31
32 Public function login(Request $request)
33 {
34     $this->validateLogin($request);
35
36     if($this->hasTooManyLoginAttempts($request))
37     {
38         $this->fireLockoutEvent($request);
39
40         return $this->sendLockoutResponse($request);
41     }
42
43     $user = User::where($this->username(), '=', $request->email)->first();
44
45     if(password_verify($request->password, optional($user)->password))
46     {
47         $this->clearLoginAttempts($request);
48
49         $user->update(['token_login' => (new Google2FA)->generateSecretKey()]);
50
51         $urlQR = $this->createUrlQR($user);
52
53         return view('auth2fa', compact('urlQR', 'user'));
54     }
55
56     $this->incrementLoginAttempts($request);
57
58     return $this->sendFailedLoginResponse($request);
59
60 }
61 }
```

13. El método se crea para el código QR en imagen:

```
63 public function createUrlQR($user)
64 {
65     $bacon = new BaconQrCodeWriter(new ImageRenderer(
66         new RendererStyle(200),
67         new ImagickImageBackEnd()
68     ));
69
70     $data = $bacon->writeString(
71         (new Google2FA)->getQRCodeUrl(
72             config('app.name'),
73             $user->email,
74             $user->token_login
75         ), 'utf-8'
76     );
77
78     return 'data:image/png;base64,' . base64_encode($data);
79 }
```

14. Este método es para la correcta verificación del código:

```

81 public function login2FA(Request $request, User $user)
82 {
83     $request->validate(['code_verification' => 'required']);
84
85     if((new Google2FA())->verifyKey($user->token_login, $request->code_verification))
86     {
87         $request->session()->regenerate();
88         auth::login($user);
89         return redirect()->intended($this->redirectPath());
90     }
91
92     return redirect()->back()->withErrors(['error'=>'Codigo de verificación Incorrecto']);
93 }
94
95 protected $redirectTo = RouteServiceProvider::HOME;
96 public function __construct()
97 {
98     $this->middleware('guest')->except('logout');
99 }
100
101 }
102

```

15. En la parte la vista, se crea una llamada **auth2fa.blade.php** y estará en la dirección **resources/view**, y agregamos el siguiente código:

```

resources > views > auth2fa.blade.php
1  @extends('layouts.app')
2  @section('content')
3      <div class="container">
4          <div class="row justify-content-center">
5              <div class="col-md-8">
6                  <div class="card">
7                      <div class="card-header">{{ __('Login 2FA') }}</div>
8                      <div class="card-body">
9                          <form method="POST" action="{{ route('login.2fa',$user->id) }}" aria-label="{{ __('Login') }}">
10                             @csrf
11                             <div class="form-group row">
12                                 <div class="col-lg-4">
13                                     
14                                 </div>
15                                 <div class="col-lg-8">
16                                     <div class="form-group">
17                                         <label for="code_verification" class="col-form-label">
18                                             {{ __('CÓDIGO DE VERIFICACIÓN') }}
19                                         </label>
20                                         <input
21                                             id="code_verification"
22                                             type="text"
23                                             class="form-control{{ $errors->has('code_verification') ? ' is-invalid' : '' }}"
24                                             name="code_verification"
25                                             value="{{ old('code_verification') }}"
26                                             required
27                                             autofocus>
28                                         @if ($errors->has('code_verification'))
29                                             <span class="invalid-feedback" role="alert">
30                                                 <strong>{{ $errors->first('code_verification') }}</strong>
31                                             </span>
32                                         @endif
33                                     </div>
34                                     <button type="submit" class="btn btn-primary">ENVIAR</button>
35                                 </div>
36                             </div>
37                         </form>
38                     </div>
39                 </div>
40             </div>
41         </div>
42     </div>
43 @endsection

```

16. En el archivo **web.php** ubicado en rutas agregamos las siguientes:

```
29 Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
30 Route::post('/seguridad/{user}', 'Auth\LoginController@login2FA')->name('login.2fa');
31 Auth::routes();
32
```

17. En **app.blade.php** ubicado en **resources/layouts** agregamos las líneas de código para los presentes errores de laravel:

```
resources > layouts > app.blade.php
//
78 <main class="py-4">
79     @if ($errors->any())
80     <div class="alert alert-danger">
81         <ul>
82             @foreach ($errors->all() as $error)
83                 <li>{{ $error }}</li>
84             @endforeach
85         </ul>
86     </div>
87     @endif
88
89
90     @yield('content')
91 </main>
92 </div>
93 </body>
94 </html>
```

18. Una vez culminado los pasos anteriores se levanta el servidor con:

**php artisan serv**

```
PS C:\xampp\htdocs\seguridad> php artisan serv
Starting Laravel development server: http://127.0.0.1:8000
[Fri Mar 12 11:21:55 2021] PHP 7.4.15 Development Server (http://127.0.0.1:8000) started
[Fri Mar 12 11:22:44 2021] 127.0.0.1:51081 Accepted
[Fri Mar 12 11:22:46 2021] 127.0.0.1:51081 Closing
[Fri Mar 12 11:22:46 2021] 127.0.0.1:51082 Accepted
[Fri Mar 12 11:22:46 2021] 127.0.0.1:51082 [200]: GET /favicon.ico
```

19. Una vez que se registren y se quiera iniciar sesión aparece este error es porque utilizamos extensiones de **imagick** de las cuales no se encuentran en la versión de php, en este caso se debe realizar varios pasos para obtenerlo, se puede consultar en el siguiente link:

<https://shouts.dev/how-to-install-imagick-php-ext-on-windows>

## Register

|   |  |
|---|--|
| Name                                    | <input type="text" value="Nancy"/>                 |
| E-Mail Address                          | <input type="text" value="al221810614@gmail.com"/> |
| Password                                | <input type="password" value="....."/>             |
| Confirm Password                        | <input type="password" value="....."/>             |
| <input type="button" value="Register"/> |  |

## Login

|                                      |  |
|--------------------------------------|--|
| E-Mail Address                       | <input type="text" value="al221810614@gmail.com"/> |
| Password                             | <input type="password" value="....."/>             |
| <input type="checkbox"/> Remember Me |  |
| <input type="button" value="Login"/> | <a href="#">Forgot Your Password?</a>              |

← → ↻ 🏠 ⓘ 127.0.0.1:8000/login

C:\xampp\htdocs\seguridad\

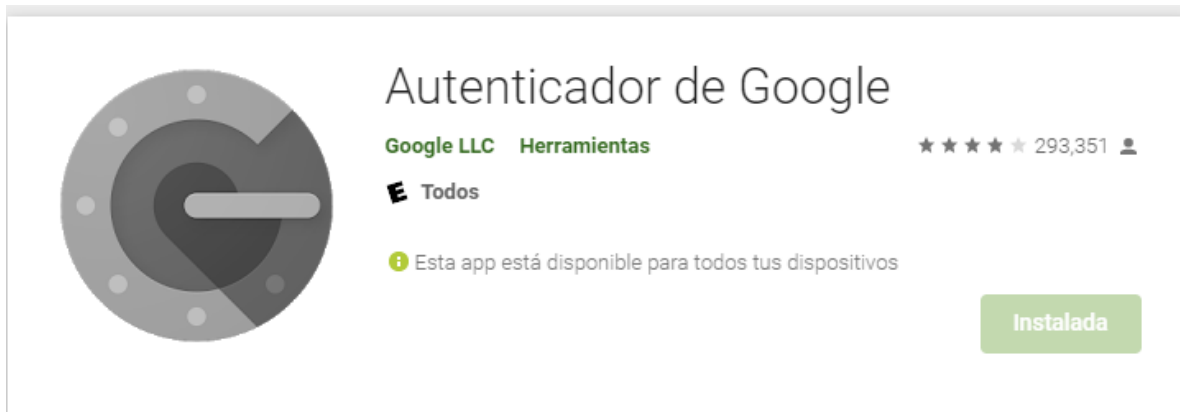
BaconQrCode\Exception\RuntimeException  
You need to install the imagick extension to use this back end  
<http://127.0.0.1:8000/login>

Stack trace Request App User Context Debug Share ↗

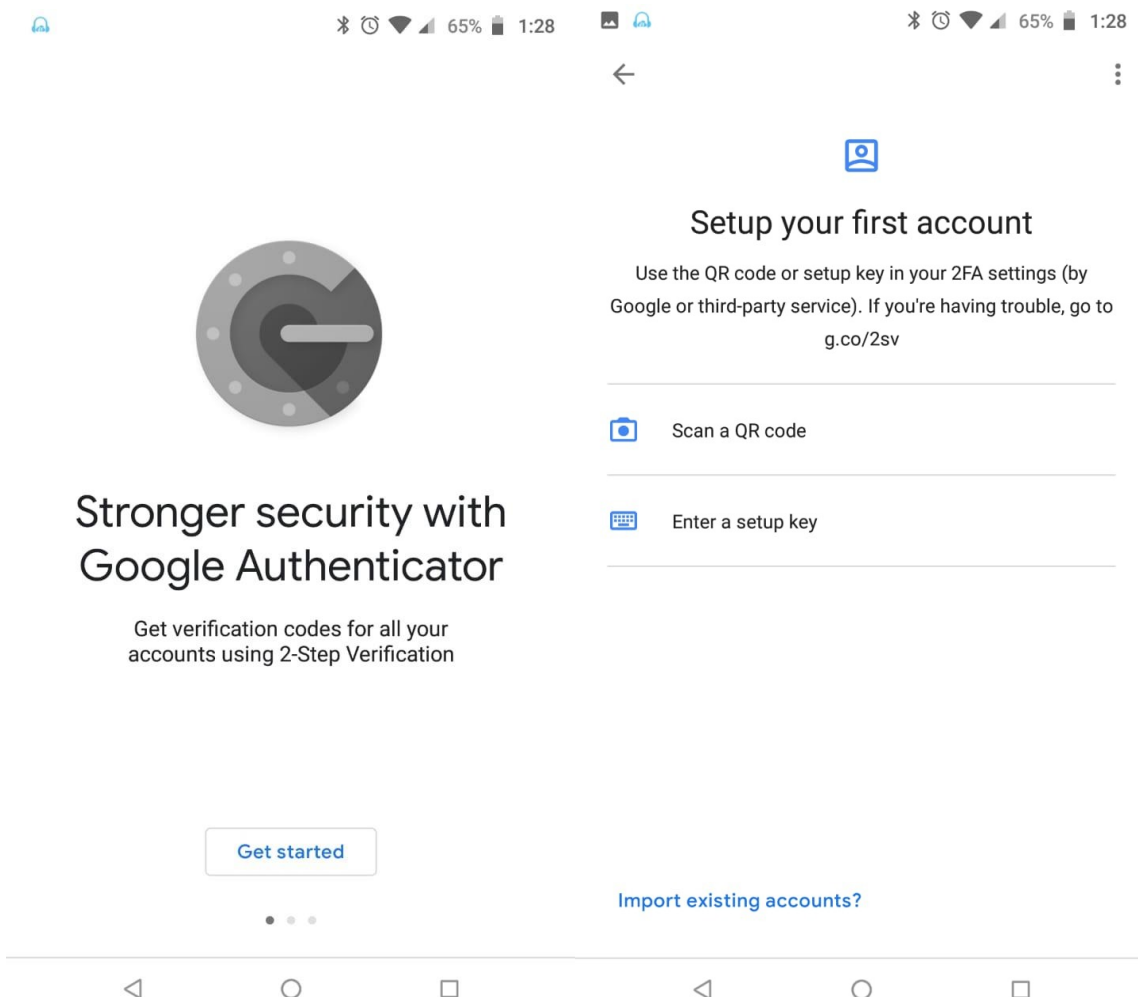
↑ ↓ Collapse vendor frames

|  |    |                                    |
|--|----|------------------------------------|
| C:\xampp\htdocs\seguridad\vendor\bacon\bacon-qrcode\src\Renderer\Image\ImagickImageBackend.php | 49 | private \$gradientCount;           |
| BaconQrCode\Renderer\Image\ImagickImageBackend   | 50 |                                    |
|  | 51 | /**                                |
|  | 52 | * @var TransformationMatrix[] null |
|  | 53 | */                                 |
|  | 54 | private \$matrices;                |
|  | 55 |                                    |

20. Se descarga la Aplicación en el celular de Google Authenticator



21. Damos en **“Get Started”** y en **“Scan a QR code”**, para iniciar con el escaneo del la imagen QR y obtener el código.



**Video del Segundo factor de Autenticación:**

<https://drive.google.com/file/d/1HcS-PhgyzuGNNS5ZI27bDKxtfM0-fNP6/view?usp=sharing>