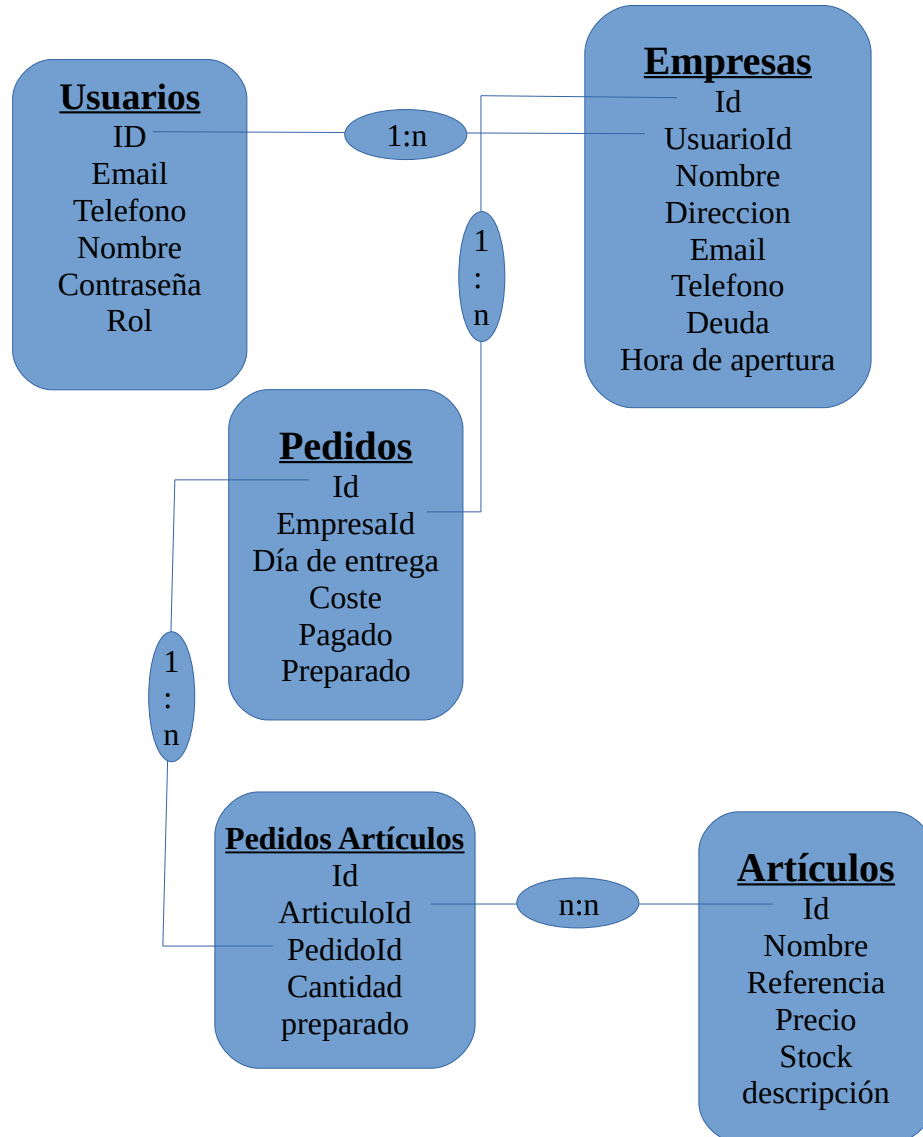


Pagina de venta a empresas

Base de datos:



*Rol: cliente o administrador

*Los preparados (booleano) de pedidos artículos era para tener funcionalidad al hacer los pedidos con los empleados en un futuro.

Las rutas

```

<?php

use ...

Route::get('/', function() {
    return view( view: 'welcome');
})->name('welcome');

//register

Route::get('register', 'Auth\RegisterController@create' )->name('register.create');
Route::post('register.create', 'Auth\RegisterController@store')->name('register.store');

//login

Route::get('form.login', 'Auth\LoginController@index')->name('form.login');
Route::post('login', 'Auth\LoginController@getin')->name('login');
Route::get('logout', 'Auth\LoginController@logout')->name('logout');

//-----client-----//

Route::group([
    'middleware' => ['auth','rol:client']
], function () {

    //crud company

    Route::get('company', 'CompanyController@index')->name('company');
    Route::get('company/create', 'CompanyController@create')->name('company.create');
    Route::post('company/store', 'CompanyController@store')->name('company.store');
    Route::get('company/change/{id}', 'CompanyController@change')->name('company.change');
    Route::patch('company/update/{id}', 'CompanyController@update')->name('company.update');
    Route::delete('company/delete/{id}', 'CompanyController@delete')->name('company.delete');
    Route::get('company/debt/{id}', 'CompanyController@debt')->name('company.debt');

```

```

//crud orders

Route::get('orders/{companyId}', 'OrdersController@index')->name('orders');
Route::get('orders/show/{id}', 'OrdersController@show')->name('orders.show');
Route::get('orders/create/{companyId}', 'OrdersController@create')->name('orders.create');
Route::post('orders/store/{companyId}', 'OrdersController@store')->name('orders.store');
Route::get('orders/change/{id}', 'OrdersController@change')->name('orders.change');
Route::patch('orders/update/{id}', 'OrdersController@update')->name('orders.update');
Route::delete('orders/delete/{id}/{companyId}', 'OrdersController@delete')->name('orders.delete');

// ordersArticle

Route::delete('ordersArticles/delete/{id}/{ordersId}', 'OrdersArticlesController@delete')->name('ordersArticles.delete');
Route::patch('ordersArticles/plusLess/{id}/{number}/{ordersId}/{operation}', 'OrdersArticlesController@plusLess')->name('ordersArticles.plusLess');
Route::get('ordersArticles/create/{ordersId}', 'OrdersArticlesController@create')->name('ordersArticles.create');
Route::post('ordersArticles/store/{articleId}/{ordersId}', 'OrdersArticlesController@store')->name('ordersArticles.store');

//PDF

Route::get('pdf', 'PdfController@getIndex');
Route::get('pdf/generar/{orderId}', 'PdfController@getGenerar');

//Router::get('pagination/{articles}/{ordersId}', 'ArticleController@pagination')->name('article.pagination');

Route::post('search/{ordersId}/{page}', 'ArticleController@search')->name('search');

});

```

```
//-----admin-----//

Route::group([
    'middleware' => ['auth','rol:admin'],
    'namespace' => 'Admin',
    'prefix' => 'admin'
], function () {

    Route::get('admin', function(){
        return view( View: 'admin.welcome');
    })->name('admin.welcome');

    //crud companies
    Route::post('companies/store', 'CompanyController@store')->name('admin.companies.store');
    Route::get('companies/create', 'CompanyController@create')->name('admin.companies.create');
    Route::get('companies', 'CompanyController@index')->name('admin.companies.show');
    Route::get('companies/change/{id}', 'CompanyController@change')->name('admin.companies.change');
    Route::patch('companies/update/{id}', 'CompanyController@update')->name('admin.companies.update');
    Route::delete('companies/delete/{id}', 'CompanyController@delete')->name('admin.companies.delete');
    Route::post('companies/restore/{id}', 'CompanyController@restore')->name('admin.companies.restore');

    //crud articles

    Route::post('articles/store', 'ArticleController@store')->name('admin.articles.store');
    Route::get('articles/create', 'ArticleController@create')->name('admin.articles.create');
    Route::get('articles', 'ArticleController@index')->name('admin.articles.show');
    Route::get('articles/change/{id}', 'ArticleController@change')->name('admin.articles.change');
    Route::patch('articles/update/{id}', 'ArticleController@update')->name('admin.articles.update');
    Route::delete('articles/delete/{id}', 'ArticleController@delete')->name('admin.articles.delete');
    Route::post('articles/restore/{id}', 'ArticleController@restore')->name('admin.articles.restore');
});
```

```
//crud orders

Route::post('orders/store', 'OrdersController@store')->name('admin.orders.store');
Route::get('orders/create', 'OrdersController@create')->name('admin.orders.create');
Route::get('orders', 'OrdersController@index')->name('admin.orders.show');
Route::get('orders/change/{id}', 'OrdersController@change')->name('admin.orders.change');
Route::post('orders/update/{id}', 'OrdersController@update')->name('admin.orders.update');
Route::delete('orders/delete/{id}', 'OrdersController@delete')->name('admin.orders.delete');
Route::post('orders/restore/{id}', 'OrdersController@restore')->name('admin.orders.restore');

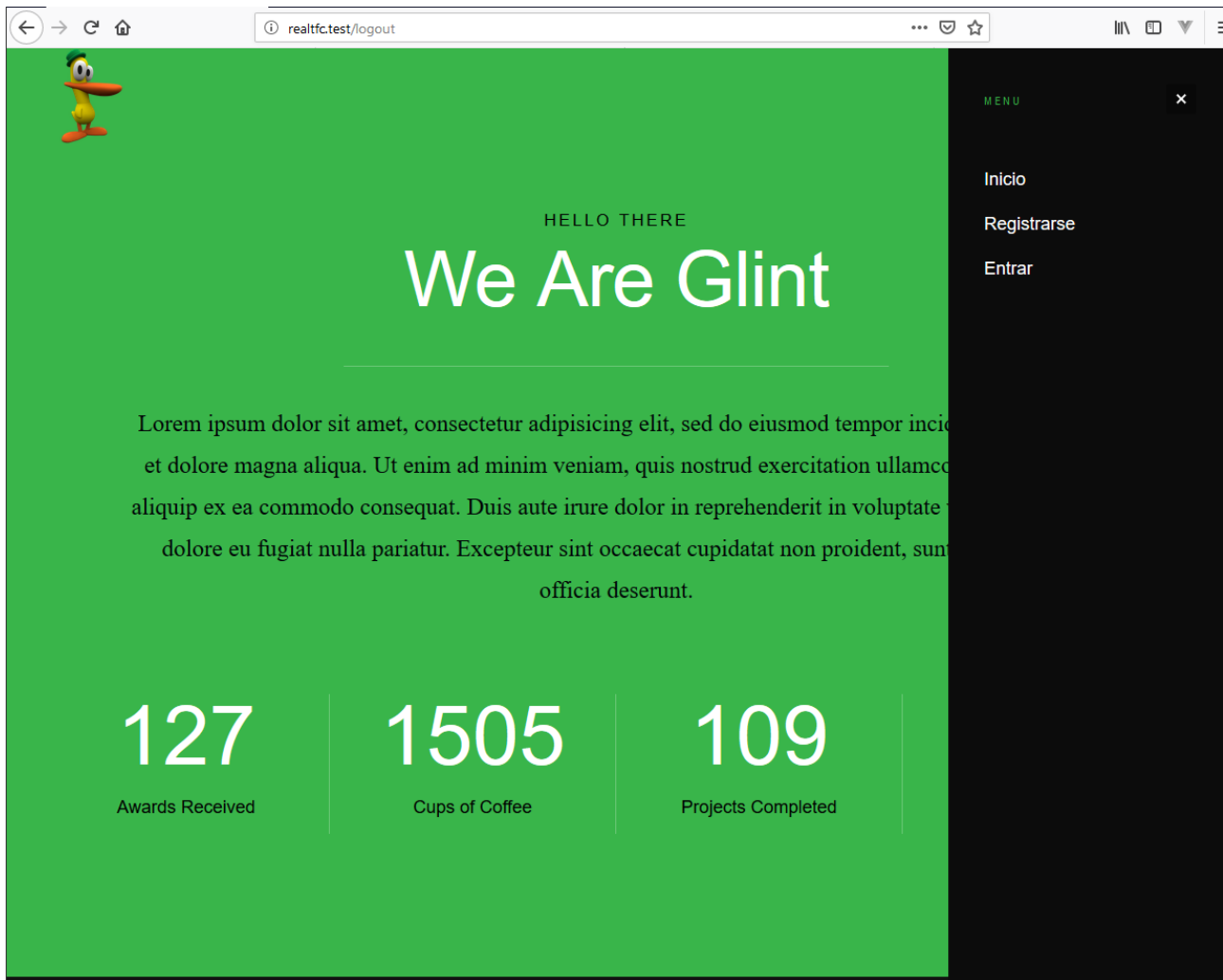
//crud ordersArticles

Route::post('ordersArticles/create/{orderId}', 'OrdersArticlesController@create')->name('admin.ordersArticles.create');
Route::post('ordersArticles/store/{orderId}', 'OrdersArticlesController@store')->name('admin.ordersArticles.store');
Route::post('ordersArticles/change/{id}', 'OrdersArticlesController@change')->name('admin.ordersArticles.change');
Route::patch('ordersArticles/update/{id}', 'OrdersArticlesController@update')->name('admin.ordersArticles.update');
Route::delete('ordersArticles/delete/{id}', 'OrdersArticlesController@delete')->name('admin.ordersArticles.delete');
Route::post('ordersArticles/restore/{id}', 'OrdersArticlesController@restore')->name('admin.ordersArticles.restore');

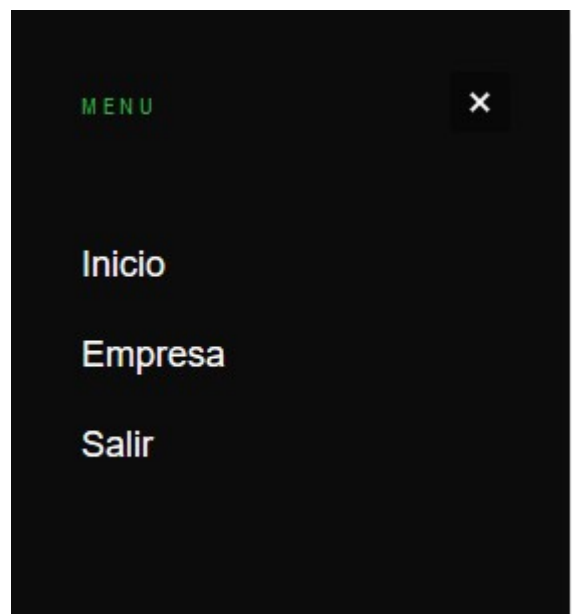
//crud users

Route::post('users/store', 'UsersController@store')->name('admin.users.store');
Route::get('users/create', 'UsersController@create')->name('admin.users.create');
Route::get('users', 'UsersController@index')->name('admin.users.show');
Route::get('users/change/{id}', 'UsersController@change')->name('admin.users.change');
Route::patch('users/update/{id}', 'UsersController@update')->name('admin.users.update');
Route::delete('users/delete/{id}', 'UsersController@delete')->name('admin.users.delete');
Route::post('users/restore/{id}', 'UsersController@restore')->name('admin.users.restore');
});
```

Inicio



Pagina inicial donde se presentaría la empresa.
Menú desplegable arriba a la derecha que enlaza a la
pagina de inicio, a registrarse y a entrar. Y si estas
dentro, a empresas y a salir.



Formulario de registro

← → ↺ 🏠

realftc.test/register

REGISTRARSE

MENU

RELLENE LOS CAMPOS

The name must be at least 3 characters.

e

NOMBRE:

The email must be a valid email address.

fdfdsa

E-MAIL:

The telephone must be at least 100000000.

456

TELEFONO:

CONTRASEÑA:

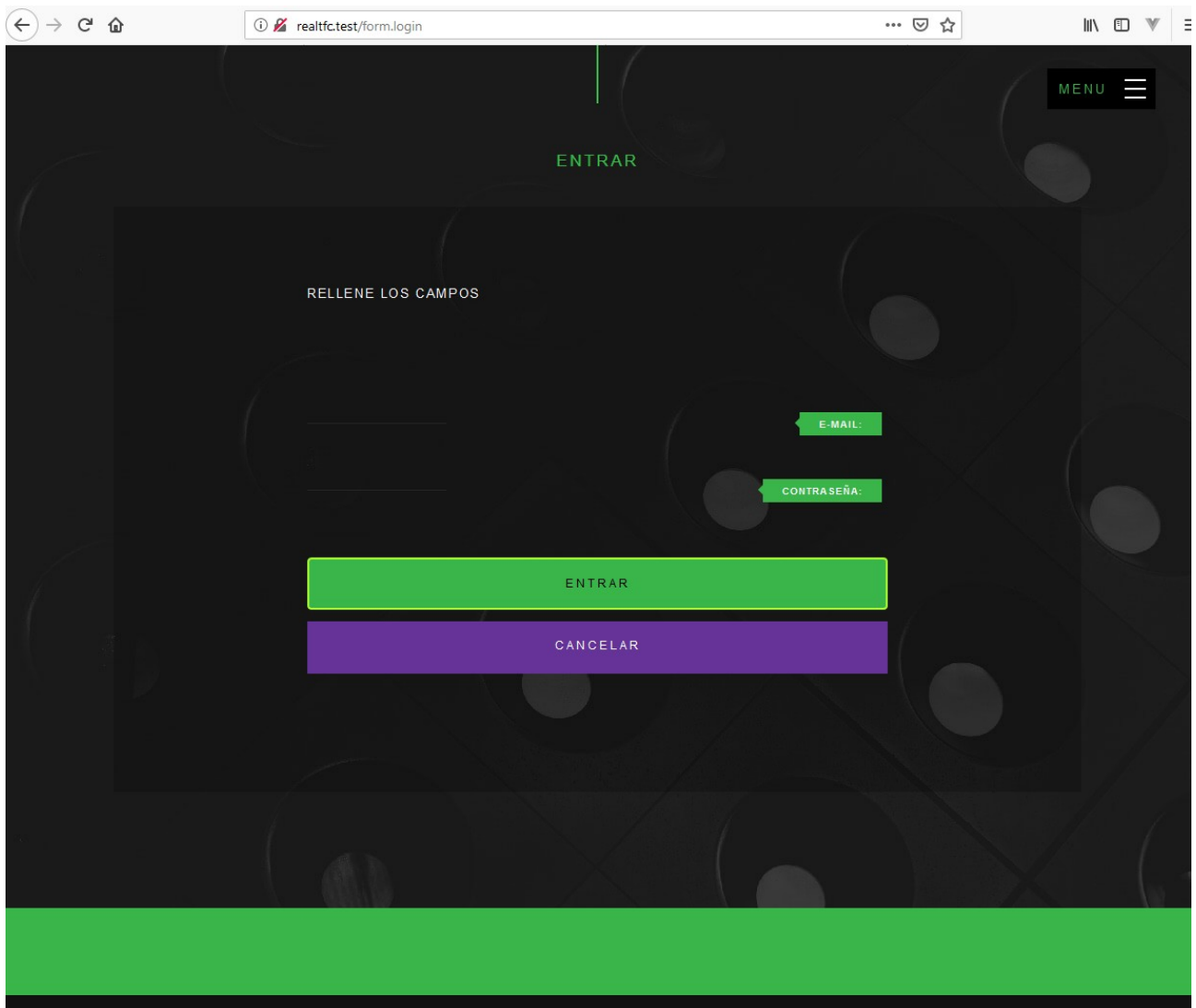
The password confirmation and password must match.

CONFIRMAR CONTRASEÑA:

GUARDAR

CANCELAR

Formulario de entrar



The image shows a web browser window with the address bar displaying 'realitfc.test/form.login'. The page has a dark background with a subtle pattern of overlapping circles. At the top right, there is a 'MENU' button with a hamburger icon. Below this, the word 'ENTRAR' is displayed in green. The main content area is a light gray rectangle containing the text 'RELLENE LOS CAMPOS' (Fill in the fields). Below this text are two input fields: 'E-MAIL:' and 'CONTRASEÑA:'. At the bottom of the form are two buttons: 'ENTRAR' (green) and 'CANCELAR' (purple). A solid green horizontal bar is at the very bottom of the page.

realitfc.test/form.login

MENU

ENTRAR

RELLENE LOS CAMPOS

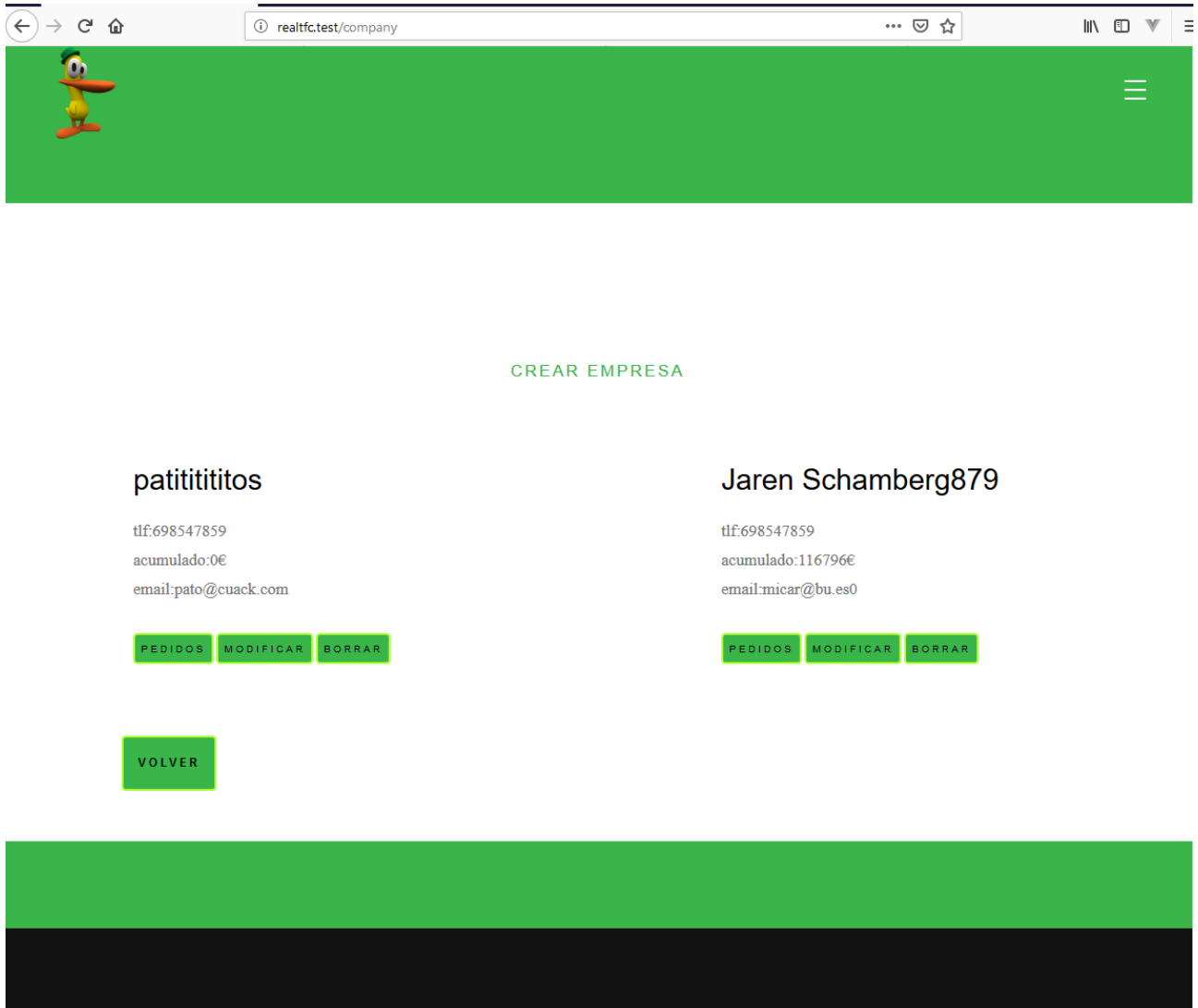
E-MAIL:

CONTRASEÑA:

ENTRAR

CANCELAR

Una vez dentro tienes en empresa



te lista las empresas, con su crud y pedidos

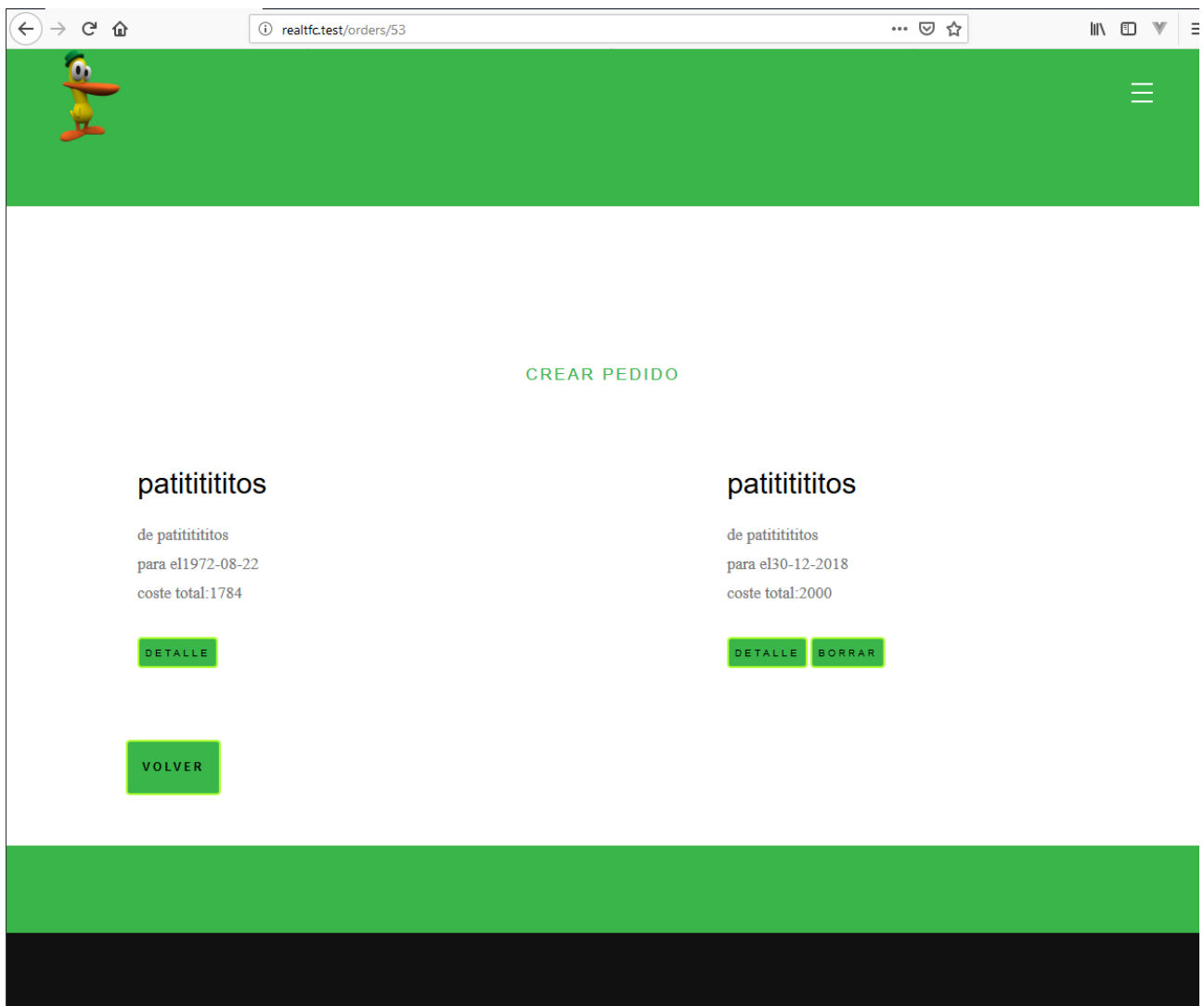
Para calcular la deuda de una empresa

```
static function calcDebt ($id)
{
    $companyDebt = Company::where('companies.id','=', $id)->join('orders', 'companies.id','=', 'orders.companyId')
        ->where('orders.charged','=', false)
        ->where('orders.open','=', false)
        ->select(DB::raw( 'SUM(orders.total) as debt'))->first();
    $debt = $companyDebt->debt;

    $company = Company::find($id);
    $company->debt = $debt;
    $company->save();

    return true;
}
```

Pedidos



realtime.test/orders/53

CREAR PEDIDO

patititititos

de patititititos
para el1972-08-22
coste total:1784

DETALLE

patititititos

de patititititos
para el30-12-2018
coste total:2000

DETALLE BORRAR

VOLVER

te muestra listado de pedidos con su crud (solo se pueden borrar si el pedido no ha sido guardado) y dentro de cada pedido

Pedido



DE: patitititos

PARA EL: 30-12-2018

CANCELAR PEDIDO

ID	NOMBRE	PEDIDOS	STOCK	PRECIO	TOTAL	MODIFICAR
131	Laurine Keeling	40	88760	50 €	2000 €	+ - BORRAR
208	Aimee Marquardt DVM	173	66000	47 €	8131 €	+ - BORRAR
					COSTE TOTAL:	10131 €

AGREGAR ARTICULO

GUARDAR

VOLVER

sumar, restar borrar o añadir articulos
se puede cancelar el pedido o guardarlo.

Para seleccionar los articulos de cada pedido, tanto para administrador como para cliente

```
static function getArticles($id, $delete=false)
{
    if($delete){
        $ordersArticles = OrdersArticles::withTrashed()->where('orderId', $id)
        ->join('articles','orders_articles.articleId','=','articles.id')
        ->select('*', 'orders_articles.id as id',
            'orders_articles.deleted_at as deleted_at',
            'orders_articles.updated_at as updated_at',
            'orders_articles.created_at as created_at')->get();
    }else{
        $ordersArticles = OrdersArticles::where('orderId', $id)
        ->join('articles','orders_articles.articleId','=','articles.id')
        ->select('*', 'orders_articles.id as id',
            'orders_articles.deleted_at as deleted_at',
            'orders_articles.updated_at as updated_at',
            'orders_articles.created_at as created_at')->get();
    }

    return $ordersArticles;
}
```

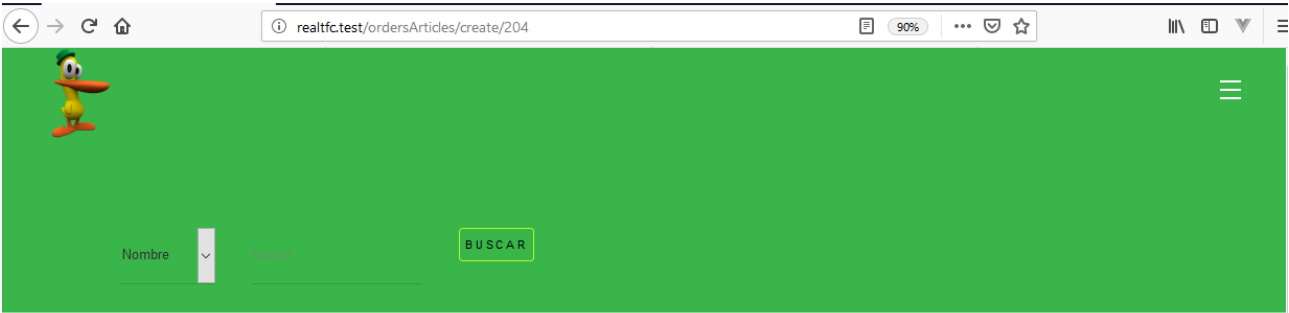
para calcular el coste total de un pedido

```
static function calcTotal($id)
{
    $order = Orders::where('orders.id', $id)
        ->join('orders_articles','orders.id','=','orders_articles.orderId')
        ->join('articles','orders_articles.articleId','=','articles.id')
        ->select(DB::raw('value: SUM(orders_articles.number*articles.price) as total'))->first();
    $total = $order->total;

    $orders = Orders::find($id);
    $orders->total = $total;
    $orders->save();

    return true;
}
```

Agregar artículos



WHAT WE DO

We've got everything you need to launch
and grow your business

Abelardo Little Sr.

Recusandae quis est molestiae officia eos aut sed iure. Aliquid necessitatibus maxime recusandae voluptates ex natus. Cum necessitatibus quos earum. Omnis dignissimos est sunt modi voluptatem.
Precio:43€
Disponibilidad:7000Kg

Cantidad:

en kg

Ada Kuvalis

Pariatur doloribus impedit cupiditate reiciendis. Et sequi in iusto doloribus sed sit nemo. Ipsam in sunt sunt sint iusto quibusdam.
Precio:34€
Disponibilidad:98000Kg

Cantidad:

en kg

Anabelle Hammes

Dolores veritatis et deserunt vero enim. Ut deserunt eum ratione expedita aperiam eaque. Quasi delectus necessitatibus animi repellat.

Precio:6€

Disponibilidad:9429Kg

Cantidad:

GUARDAR

en kg

Angelo Conn

Distinctio voluptas dignissimos aut et et. Quod id et dolorem commodi et. Qui ea possimus reprehenderit placeat qui. Fuga quibusdam quo vel similique.

Precio:22€

Disponibilidad:35016Kg

Cantidad:

GUARDAR

en kg

< 1 2 3 4 5 6 7 8 ... 16 17 >

te lista los artículos con su paginación y con un buscador en la parte de arriba que puede buscar tanto por título o parte de este como por descripción para calcular el stock de los artículos

```
static function stockcalc($articleId, $number, $operation)
{
    if($operation=='plus'){
        $article = Article::find($articleId);
        $article->stock += $number;
    }
    if($operation=='less'){
        $article = Article::find($articleId);
        $article->stock -= $number;
    }
    $article->save();
}
```

Pedido

DE: patitititos

PARA EL: 30-12-2018

ID	NOMBRE	PEDIDOS	PRECIO	TOTAL	MODIFICAR
131	Laurine Keeling	40	50 €	2000 €	--
208	Aimee Marquardt DVM	173	47 €	8131 €	--
209	Alysha Lubowitz	45	14 €	630 €	--
				COSTE TOTAL:	10761 €

VER FACTURA

DESCARGAR FACTURA

VOLVER

una vez guardado te da la opcion de ver o descargar una factura en pdf del pedido.



Jr. Santander Nro. 340 Jesus María - Lima

Telf.: (01) 364-2547 Cel.: 985-748514

RUC: 10072486893

BOLETA DE VENTA

001- N° 00000412

Señor (es): patitititos

Dirección: 428 Aniyah LodgeSashaside, KY 18866-8604

Tlf: 698547859

DÍA	MES	AÑO
16	12	2018

CANT.	NOMBRE	PRECIO	IMPORTE
40 Kg	Laurine Keeling	50 €	2000 €
173 Kg	Aimee Marquardt DVM	47 €	8131 €
45 Kg	Alysha Lubowitz	14 €	630 €
		TOTAL S/.	10 761.00 €

CANCELADO

EMISOR

Documentacion en <http://desarrollowebtutorial.com/generar-pdf-en-laravel/>

```

<?php

namespace App\Http\Controllers;

use ...

class PdfController extends Controller
{
    public function getIndex()
    {
        return view( view: 'pdf.index');
    }
    public function getGenerar(Request $request, $orderId)
    {
        $accion = $request->get( key: 'accion');
        $tipo = $request->get( key: 'tipo');
        Pdf::pdf($accion,$tipo,$orderId);
    }
}

```

cuando guardas el pedido te llega un email a tu correo de la empresa avisándote de que el pedido esta guardado

```

static function orderSave($orders)
{
    $company = Company::find($orders->companyId);
    $user = User::find($company->userId);
    $data = [
        'companyName' => $company->name,
        'orders' => $orders->id,
        'email'=> $company->email,
        'userName' => $user->name,
    ];

    Mail::send( view: 'emails.notification', $data, function($msg) use ($data) {
        $msg->from('c4c0t4s@gmail.com', 'Pato Cuack');
        $msg->to($data['email'])->subject('Pedido realizado');
    });
}

```

```

MAIL_DRIVER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME='c4c0t4s@gmail.com'
MAIL_PASSWORD=*****
MAIL_ENCRYPTION=tls

```

Pedido realizado



Pato Cuack <c4c0t4s@gmail.com>

Sáb 08/12/2018, 18:27

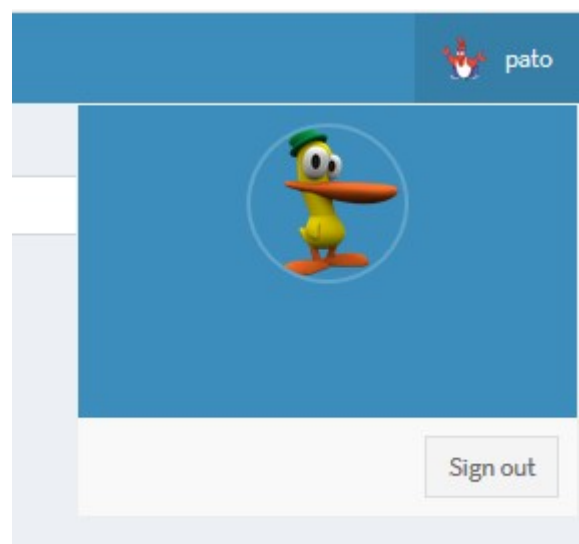
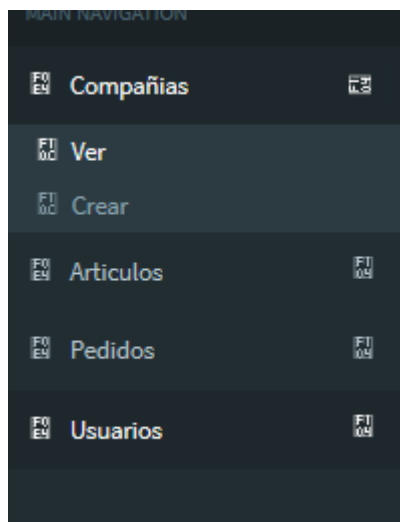
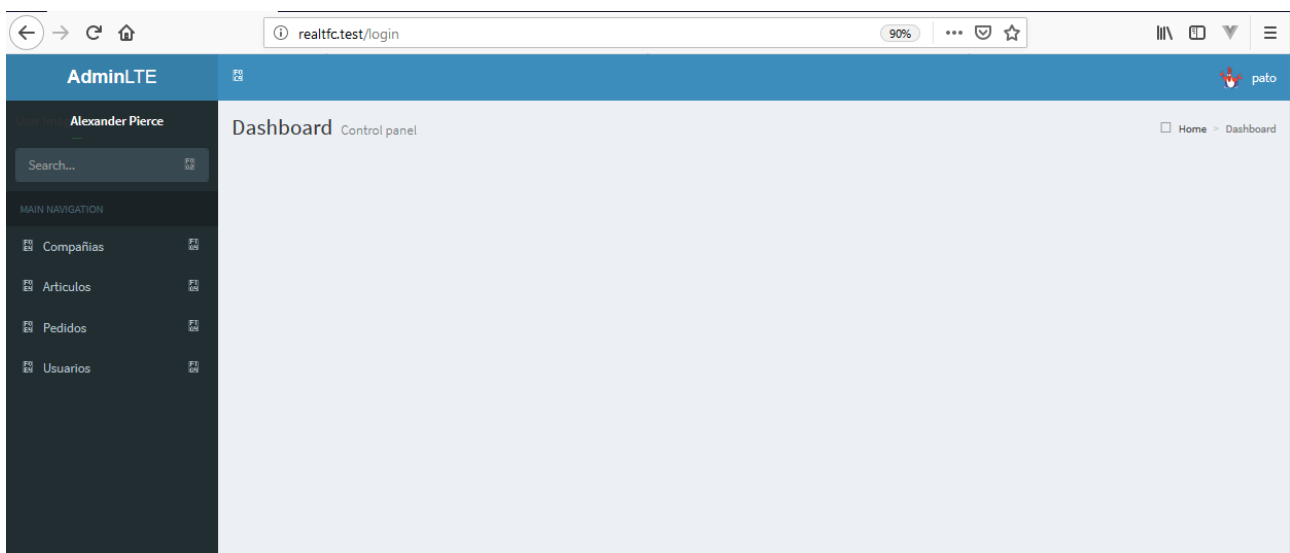
Usted ▾

Buenos días munar.

El pedido 206 de cacotas ha sido guardado

Administración

si tu rol es admin al entrar te lleva al panel de administración



las cuatro opciones del menú tienen la opción de listar con paginacion(donde esta el crud y la recuperación de borrados) y crear.

En la creación de pedidos esta el crud y creación de artículos en el pedido(pedidos-artículos)

The screenshot shows the AdminLTE interface. The sidebar on the left contains the user name 'Alexander Pierce' and a search bar. Below it is the 'MAIN NAVIGATION' section with links to 'Compañías', 'Artículos', 'Pedidos', and 'Usuarios'. The main content area is titled 'Data Tables' and contains a 'Pedido' form and a table of existing orders.

Form Fields:

- Compañía: patitititos
- Coste: 10761
- Fecha de entrega: 30-12-2018
- Cobrado: no
- Pedido abierto: no

Table: Pedido

id	articleId	orderId	number	prepare	deleted_at	created_at	updated_at	name	reference	price	stock	description	Modificar
131	179	204	40	1		2018-12-14 14:50:54	2018-12-15 20:33:46	Laurine Keeling	3287	50	88760	Quia expedita expedita repellendus itaque at qui. Nihil dignissimos odio molestias dolorem.	Detalle Borrar
208	77	204	173	0		2018-12-16 16:49:07	2018-12-16 16:49:07	Aimee Marquardt DVM	3687	47	66000	Iusto est est ipsam totam voluptatem minus. Sed sed distinctio autem nobis non vero. Unde eum aut sit cum similique placeat rerum.	Detalle Borrar
209	84	204	45	0	2018-12-16 17:22:50	2018-12-16 17:05:58	2018-12-16 17:22:50	Alysha Lubowitz	5141	14	57436	Vero dolore voluptatem accusamus dolorem nobis quia. Numquam cum omnis sed voluptas laudantium aperiam et. Nulla tempora sequi quo inventore repudiandae.	Resataurar

*Toda entrada de datos tiene sus validaciones con sus “requests” y pasa por laravel collective por seguridad.

*Cualquier borrado es solo parcial(soft delete).

Paginación antes de descubrir la de laravel

```
static function paginate($table,$camp,$data,$showMany=10,$page=1)
{
    $skip = $showMany*$page- $showMany;
    $items = DB::table($table)->where($camp, operator: 'LIKE', value: '%'.$data.'%')->take($showMany)->skip($skip)->get();
    $count= DB::table($table)->where($camp, operator: 'LIKE', value: '%'.$data.'%')->count();
    if ($count+$showMany-1 < $showMany*$page){
        return false;
    }

    return ['count'=>$count,'items'=>$items];
}
```

reglas personalizadas

MinTomorrow

```

class MinTomorrow implements Rule
{
    /**
     * Create a new rule instance.
     *
     * @return void
     */
    public function __construct()
    {
        //
    }

    /**
     * Determine if the validation rule passes.
     *
     * @param string $attribute
     * @param mixed $value
     * @return bool
     */
    public function passes($attribute, $value)
    {
        if (Carbon::createFromFormat( format: 'd-m-Y', $value)){
            if (Carbon::createFromFormat( format: 'd-m-Y', $value) <= Carbon::now()) {
                return false;
            }else{
                return true;
            }
        }else{
            return false;
        }
    }
}

```

emailCompanyUpdate

```

class EmailCompanyUpdate implements Rule
{
    /**
     * Create a new rule instance.
     *
     * @return void
     */
    public function __construct()
    {
        //
    }

    /**
     * Determine if the validation rule passes.
     *
     * @param string $attribute
     * @param mixed $value
     * @return bool
     */
    public function passes($attribute, $value)
    {
        if (count(Company::where('email', $value)->get()) > 0) {
            if (Company::where('userId', auth()->id())->where('email', $value)->first()) {
                return true;
            }else{
                return false;
            }
        }else{
            return true;
        }
    }
}

```