



# Manual del Desarrollador - Selfinance

## 1 Introducción

Este documento describe la arquitectura, configuración y desarrollo de **Selfinance**, una plataforma de gestión de finanzas personales desarrollada con **Laravel**.

## 2 Requisitos del Sistema



### Software Necesario:

- PHP >= 8.2.12
- Composer >= 2.8.4
- MySQL o PostgreSQL
- Node.js y NPM (para assets frontend)
- Git (para control de versiones)

## 3 Estructura del Proyecto

- 📁 **app/** → Contiene los controladores, modelos y lógica del negocio.
- 📁 **database/** → Migraciones, factories y seeders.
- 📁 **routes/** → Rutas de la aplicación.
- 📁 **resources/** → Vistas Blade y archivos CSS/JS.
- 📁 **public/** → Archivos estáticos y punto de entrada.
- 📁 **config/** → Archivos de configuración.

## 4 Gestión de Usuarios y Autenticación

### ♦ Roles de Usuario

Actualmente no existen roles de usuario.

### ♦ Middleware de Autenticación y Manejo de Rutas

Se usa **auth** para la autenticación del usuario.

```
Route::get(uri: '/', action: function (): mixed {
    return view(view: 'welcome');
});

Auth::routes();

Route::get(uri: '/home', action: [App\Http\Controllers\HomeController::class, 'index'])->name('home');

Route::resource(name: 'transactions', controller: TransactionController::class)->middleware('auth');
Route::resource(name: 'clients', controller: ClientController::class)->middleware('auth');

use App\Http\Controllers\PageController;

Route::get(uri: '/about', action: [PageController::class, 'about'])->name('about');
Route::get(uri: '/contact', action: [PageController::class, 'contact'])->name('contact');
```

## 5 .env

Cambio del APP\_NAME y la conexión a la base de datos

```
APP_NAME=Selfinance
APP_ENV=local
APP_KEY=base64:frKbhixsw0cX6q0mgyIjf0BiTvHffuVbQgC0PdJxwk8=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=proyecto_laravel
DB_USERNAME=root
DB_PASSWORD=
```

## 6 Base de Datos

 Estructura de la tabla **clients**

Relacionada con la tabla users, esta tabla guarda datos personales tales como el balance, el número de teléfono, etc.

```
Schema::create(table: 'clients', callback: function (Blueprint $table): void {
    $table->id();
    $table->unsignedBigInteger(column: 'user_id');
    $table->string(column: 'name')->nullable();
    $table->string(column: 'email')->nullable();
    $table->string(column: 'phone_number')->nullable();
    $table->decimal(column: 'balance', total: 10, places: 2)->default(value: 0);
    $table->timestamps();

    $table->foreign(columns: 'user_id')->references(columns: 'id')->on(table: 'users')->onDelete(action: 'cascade');
});
```

## Estructura de la tabla **transactions**

Relacionada con la tabla users, registra todas las transacciones relacionadas por los usuarios.

```
Schema::create(table: 'transactions', callback: function (Blueprint $table): void {
    $table->engine="InnoDB";
    $table->id();
    $table->bigInteger(column: 'user_id')->unsigned();
    $table->decimal(column: 'amount', total: 10, places: 2);
    $table->enum(column: 'type', allowed: ['deposit', 'withdrawal', 'transfer', 'payment']);
    $table->string(column: 'currency', length: 3)->default(value: 'EUR');
    $table->enum(column: 'status', allowed: ['pending', 'completed', 'failed', 'reversed']->default(value: 'pending');
    $table->timestamps();
    $table->foreign(columns: 'user_id')->references(columns: 'id')->on(table: 'users')->onDelete(action: 'cascade');
});
```

## **7** Seeders

### Estructura de la clase **AdminSeeder**

Seeder qué crea el usuario admin por defecto y su respectiva entrada en la clase client.

```

/**
 * Run the database seeds.
 */
0 references | 0 overrides
public function run(): void
{
    $admin = User::create(attributes: [
        'name' => 'Admin User',
        'email' => 'sfadmin@selfinance.com',
        'password' => Hash::make(value: 'adminpassword'),
    ]);

    Client::create(attributes: [
        'user_id' => $admin->id,
        'name' => $admin->name,
        'email' => $admin->email,
        'phone_number' => '000000000',
        'balance' => 0.00,
    ]);
}

```

## 8 Listeners

### Estructura de la clase **CreateClientAfterRegister**

Al registrarse un nuevo usuario le crea una entrada en la clase client a modo de perfil para más tarde editar.

```

/**
 * Handle the event.
 *
 * @param \Illuminate\Auth\Events\Registered $event
 * @return void
 */
0 references | 0 overrides
public function handle(Registered $event): void
{
    // Crear un cliente asociado al nuevo usuario registrado
    Client::create(attributes: [
        'user_id' => $event->user->id,
        'name' => $event->user->name,
        'email' => $event->user->email,
        'phone_number' => null,
        'balance' => 0,
    ]);
}

```

## 9 Modelos

### Estructura de la clase **Client**

Un cliente pertenece a un usuario y posee esos campos fillable.

```

class Client extends Model
{
    0 references
    protected $perPage = 20;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    0 references
    protected $fillable = ['user_id', 'name', 'email', 'phone_number', 'balance'];

    /**
     * @return \Illuminate\Database\Eloquent\Relations\BelongsTo
     */
    0 references | 0 overrides
    public function user(): BelongsTo
    {
        return $this->belongsTo(related: \App\Models\User::class, foreignKey: 'user_id', ownerKey: 'id');
    }
}

```

## Estructura de la clase Transaction

Una transacción pertenece a un usuario y posee esos fillable.

```

class Transaction extends Model
{
    0 references
    protected $perPage = 20;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    0 references
    protected $fillable = ['user_id', 'amount', 'type', 'currency'];

    0 references | 0 overrides
    public function user(): BelongsTo
    {
        return $this->belongsTo(related: User::class); // La transacción pertenece a un usuario
    }
}

```

## 10 Controladores

### Estructura de la clase **ClientController**

Esta clase ClientController en Laravel maneja la gestión de clientes vinculados a usuarios.

index(): Muestra el perfil del cliente logueado si existe; si no, redirige a la vista de creación de perfil.

show(): Muestra los detalles de un cliente específico.

edit(): Carga la vista para editar un cliente.

update(): Valida y actualiza la información del cliente.

destroy(\$id):

Borra el cliente y su usuario asociado.

Cierra sesión automáticamente.

Redirige a la página principal.

```
0 references | 0 overrides
public function index(): View
{
    $client = Client::where(column: 'user_id', operator: Auth::id())->first();

    if ($client) {
        return view(view: 'client.index', data: compact(var_name: 'client'));
    } else {
        return view(view: 'client.create')->with(key: 'error', value: 'Client not found. Please create your profile.');
```

```
0 references | 0 overrides
public function show($id): View
{
    $client = Client::findOrFail(id: $id);

    return view(view: 'client.show', data: compact(var_name: 'client'));
}
```

```

0 references | 0 overrides
public function edit($id): View
{
    $client = Client::find(id: $id);

    $user = User::pluck(column: 'email', key: 'id');

    return view(view: 'client.edit', data: compact(var_name: 'client', var_names: 'user'));
}

```

```

0 references | 0 overrides
public function update(Request $request, Client $client): RedirectResponse
{
    $validated = $request->validate(rules: [
        'name' => 'required|string|max:255',
        'email' => 'required|email|max:255',
        'phone_number' => 'required|numeric',
        'balance' => 'required|numeric',
    ]);

    $client->update(values: $validated);

    return redirect()->route(route: 'clients.index')->with(key: 'success', value: 'Profile updated successfully.');
```

```

0 references | 0 overrides
public function destroy($id): RedirectResponse
{
    $client = Client::findOrFail(id: $id);

    $user = $client->user;

    $client->delete();

    if ($user) {
        $user->delete();
        Auth::logout();
    }

    return Redirect::route(route: 'home')
        ->with(key: 'success', value: 'Client and user deleted successfully, you are logged out.');
```

## Estructura de la clase TransactionController

Esta clase Transaction Controller en Laravel gestiona las transacciones financieras de los usuarios.

index():

Muestra sólo las transacciones del usuario autenticado.



Paginación de 10 transacciones por página.

create():

Muestra el formulario para crear una nueva transacción.

Carga la lista de usuarios disponibles.

store():

Valida y guarda una nueva transacción en la base de datos.

Asigna la transacción al usuario autenticado.

edit(\$id):

Muestra el formulario para editar una transacción.

Carga la lista de usuarios disponibles.

```
0 references | 0 overrides
public function index(Request $request): View
{
    $transactions = Transaction::where(column: 'user_id', operator: auth()->id())
        ->paginate(perPage: 10);

    return view(view: 'transaction.index', data: compact(var_name: 'transactions'))
        ->with(key: 'i', value: ($request->input(key: 'page', default: 1) - 1) * $transactions->perPage());
}
```

```
0 references | 0 overrides
public function create(): View
{
    $transaction = new Transaction();
    $user = User::pluck(column: 'name', key: 'id');

    return view(view: 'transaction.create', data: compact(var_name: 'transaction', var_names: 'user'));
}
```

```

0 references | 0 overrides
public function store(TransactionRequest $request): RedirectResponse
{
    $data = $request->validated();

    $data['user_id'] = auth()->id();

    Transaction::create(attributes: $data);

    return Redirect::route(route: 'transactions.index')
        ->with(key: 'success', value: 'Transaction created successfully.');
```

```

0 references | 0 overrides
public function edit($id): View
{
    $transaction = Transaction::find(id: $id);
    $user = User::pluck(column: 'name', key: 'id');

    return view(view: 'transaction.edit', data: compact(var_name: 'transaction', var_names: 'user'));
```

## 📌 Estructura de la clase **PageController**

La clase PageController en Laravel gestiona las páginas informativas accesibles para todos los usuarios.

about():

Carga y muestra la vista de la página "About" (pages.about).

contact():

Carga y muestra la vista de la página "Contact" (pages.contact).

```

3 references | 0 implementations
class PageController extends Controller
{
    /**
     * Show the About page.
     */
    1 reference | 0 overrides
    public function about(): Factory|View
    {
        return view(view: 'pages.about');
    }

    /**
     * Show the Contact page.
     */
    1 reference | 0 overrides
    public function contact(): Factory|View
    {
        return view(view: 'pages.contact');
    }
}

```

## Vistas

 Estructura de la vista **app.blade**

Enlaces a las distintas vistas, tanto estando identificado como no.

```

<!-- Left Side Of Navbar -->
<ul class="navbar-nav me-auto">
  @if (Auth::check())
    <li class="nav-item">
      <a class="nav-link" href="{{ route(name: 'clients.index') }}">{{ __(key: 'Client') }}</a>
    </li>

    <li class="nav-item">
      <a class="nav-link" href="{{ route(name: 'transactions.index') }}">{{ __(key: 'Transactions') }}</a>
    </li>
  @endif
  <li class="nav-item">
    <a class="nav-link" href="{{ route(name: 'about') }}">{{ __(key: 'About') }}</a>
  </li>

  <li class="nav-item">
    <a class="nav-link" href="{{ route(name: 'contact') }}">{{ __(key: 'Contact') }}</a>
  </li>
</ul>

```

📌 Estructura de la vista **about.blade**

```

@extends(view: 'layouts.app')

@section(section: 'content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">{{ __('key: 'About Us') }}</div>

                    <section id="about-us" class="about-section card-body">
                        <div class="container">
                            <h2>About Us</h2>
                            <p>Welcome to <strong>SelfFinance</strong>, where your financial well-being is our top priority. We specialize in providing innovative and personalized financial management solutions that help you achieve financial freedom and make informed decisions with your money.</p>

                            <h3>Our Mission</h3>
                            <p>At <strong>SelfFinance</strong>, our mission is simple: to empower individuals and families with the tools and knowledge they need to take control of their finances. We are committed to helping you build a secure financial future by offering professional guidance and innovative strategies that are tailored to your unique needs.</p>

                            <h3>What We Do</h3>
                            <p>We offer a comprehensive range of personal finance management services that include:</p>
                            <ul>
                                <li><strong>Budgeting & Planning</strong>: Helping you create and stick to a budget that aligns with your goals.</li>
                                <li><strong>Debt Management</strong>: Providing strategies to help you pay off debt efficiently.</li>
                                <li><strong>Investment Guidance</strong>: Offering expert advice on how to make smart investments.</li>
                                <li><strong>Financial Education</strong>: Empowering you with the knowledge to understand your financial situation and make better decisions.</li>
                                <li><strong>Retirement Planning</strong>: Assisting you in planning for a financially secure retirement.</li>
                            </ul>
                            <p>Our user-friendly platform makes it easier than ever to track your spending, create goals, and monitor your progress toward financial freedom.</p>

                            <h3>Our Approach</h3>
                            <p>At <strong>SelfFinance</strong>, we take a personalized approach to financial management. We understand that every client's financial situation is unique, and we are here to listen, understand, and help you take the best path forward. Whether you're looking to reduce debt, save for a big purchase, or plan for the future, we're here to provide the support you need every step of the way.</p>
                        </div>
                    </section>
                </div>
            </div>
        </div>
    </div>

```

```

<h3>Our Values</h3>
<ul>
  <li><strong>Integrity</strong>: We believe in providing honest and transparent financial
    advice.</li>
  <li><strong>Empathy</strong>: We understand that financial decisions can be overwhelming,
    and we approach each client with care and understanding.</li>
  <li><strong>Innovation</strong>: We are constantly evolving and embracing the latest tools
    and technology to make personal finance management easier for you.</li>
  <li><strong>Customer-Centric</strong>: Your success is our success. We focus on your needs
    and goals, providing solutions that work for you.</li>
</ul>

<h3>Why Choose Us?</h3>
<ul>
  <li><strong>Expertise</strong>: Our team of financial professionals has years of experience
    in managing personal finances.</li>
  <li><strong>Tailored Solutions</strong>: We understand that one size does not fit all, which
    is why we provide personalized solutions that work for you.</li>
  <li><strong>Easy-to-Use Tools</strong>: Our platform is designed with simplicity in mind,
    making financial management easier and more intuitive.</li>
  <li><strong>Customer Support</strong>: We are always here to help. Our dedicated customer
    support team is ready to answer any questions you have and provide the assistance you
    need.</li>
</ul>
</div>
</section>
</div>
</div>
</div>
@endsection

```

📌 Estructura de la vista **contact.blade**

```

@extends(view: 'layouts.app')

@section(section: 'content')
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <div class="card-header">{{ __(key: 'Contact Us') }}</div>

        <div class="card-body">
          <h2>Contact Information</h2>
          <p>If you have any questions, feel free to reach out to us.</p>
          <ul>
            <li><strong>Email:</strong> kabeCEO@selffinance.com</li>
            <li><strong>Phone:</strong> (+34) 654 321 987</li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</div>
@endsection

```

📌 Estructura de la vista **transaction.index**

```

@extends(view: 'layouts.app')

@section(section: 'template_title')
    Transactions
@endsection

@section(section: 'content')
    <div class="container-fluid">
        <div class="row">
            <div class="col-sm-12">
                <div class="card">
                    <div class="card-header">
                        <div style="display: flex; justify-content: space-between; align-items: center;">
                            <span id="card_title">
                                {{ __(key: 'Transactions') }}
                            </span>
                            <div class="float-right">
                                <a href="{{ route(name: 'transactions.create') }}" class="btn btn-primary btn-sm float-right" data-placement="left">
                                    {{ __(key: 'Create New') }}
                                </a>
                            </div>
                        </div>
                    </div>
                </div>

                @if ($message = Session::get(key: 'success'))
                    <div class="alert alert-success m-4">
                        <p>{{ $message }}</p>
                    </div>
                @endif

                <div class="card-body bg-white">
                    <div class="table-responsive">
                        <table class="table table-striped table-hover">
                            <thead class="thead">
                                <tr>
                                    <th>#</th>
                                    <th>User</th>
                                    <th>Amount</th>
                                    <th>Type</th>
                                    <th>Currency</th>
                                    <th>Status</th>
                                    <th>Date</th>
                                    <th>Actions</th>
                                </tr>
                            </thead>
                            <tbody>
                                @foreach ($transactions as $index => $transaction)
                                    <tr>

```

```

                                        <td>{{ $transaction->id }}</td>
                                        <td>{{ $transaction->user->name ?? 'sin usuario' }}</td>
                                        <td>{{ number_format(num: $transaction->amount, decimals: 2) }}</td>
                                        <td>{{ ucfirst(string: $transaction->type) }}</td>
                                        <td>{{ $transaction->currency }}</td>
                                        <td>
                                            <span class="badge bg-{{ $transaction->status == 'completed' ? 'success' : ($transaction->status == 'failed' ? 'danger' : 'warning') }}">
                                                {{ ucfirst(string: $transaction->status) }}
                                            </span>
                                        </td>
                                        <td>{{ $transaction->created_at->format('d/m/Y H:i') }}</td>
                                        <td>
                                            <form action="{{ route(name: 'transactions.destroy', parameters: $transaction->id) }}" method="POST">
                                                <!-- <a class="btn btn-sm btn-primary" href="{{ route('transactions.show', $transaction->id) }}">
                                                    <i class="fa fa-fw fa-eye"></i> {{ __(key: 'Show') }}
                                                </a> -->
                                                <a class="btn btn-sm btn-success" href="{{ route(name: 'transactions.edit', parameters: $transaction->id) }}">
                                                    <i class="fa fa-fw fa-edit"></i> {{ __(key: 'Edit') }}
                                                </a>
                                                @csrf
                                                @method('DELETE')
                                                <button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure to delete?')">
                                                    <i class="fa fa-fw fa-trash"></i> {{ __(key: 'Delete') }}
                                                </button>
                                            </form>
                                        </td>
                                    </tr>
                                @endforeach
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
        </div>
        {!! $transactions->withQueryString()->links() !!}
    </div>
@endsection

```



## Estructura de la vista **transaction.form**

```
<div class="row padding-1 p-1">
  <div class="col-md-12">
    <div class="row padding-1 p-1">
      <!-- Cantidad -->
      <div class="col-md-12">
        <label for="amount">Amount</label>
        <input type="number" step="0.01" name="amount"
          class="form-control{{ $errors->has('amount') ? ' is-invalid' : '' }}"
          value="{{ old(key: 'amount', default: $transaction->amount) }}"
          {!! $errors->first('amount', '<div class="invalid-feedback">:message</div>') !!}
        </div>
      <!-- Tipo de transacción -->
      <div class="col-md-12">
        <label for="type">Transaction type</label>
        <select name="type" class="form-control{{ $errors->has('type') ? ' is-invalid' : '' }}">
          <option value="">Select one</option>
          @foreach(['deposit' => 'Deposit', 'withdrawal' => 'Withdrawal', 'transfer' => 'Transfer', 'payment' => 'Payment'] as $key => $value)
            <option value="{{ $key }}" {{ old(key: 'type', default: $transaction->type) == $key ? 'selected' : '' }}>
              {{ $value }}
            </option>
          @endforeach
        </select>
        {!! $errors->first('type', '<div class="invalid-feedback">:message</div>') !!}
      </div>
      <!-- Moneda -->
      <div class="col-md-12">
        <label for="currency">Currency</label>
        <input type="text" name="currency"
          class="form-control{{ $errors->has('currency') ? ' is-invalid' : '' }}"
          value="{{ old(key: 'currency', default: $transaction->currency) }}"
          {!! $errors->first('currency', '<div class="invalid-feedback">:message</div>') !!}
        </div>
      <div class="col-md-12 mt20 mt-2">
        <button type="submit" class="btn btn-primary">{{ __(key: 'Submit') }}</button>
      </div>
    </div>
  </div>
</div>
```

## Estructura de la vista **client.index**

```

@extends(view: 'layouts.app')

@section(section: 'template_title')
    Client Profile
@endsection

@section(section: 'content')
    <div class="container-fluid">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">
                        <div style="display: flex; justify-content: space-between; align-items: center;">
                            <span id="card_title">
                                {{ __(key: 'Client Profile') }}
                            </span>
                            <div class="float-right">
                                <a href="{{ route(name: 'clients.edit', parameters: $client->id) }}" class="btn btn-success btn-sm float-right">
                                    {{ __(key: 'Edit Profile') }}
                                </a>
                            </div>
                        </div>
                    </div>
                    <div class="card-body">
                        <div class="row mb-3">
                            <div class="col-md-4 font-weight-bold">{{ __(key: 'User ID') }}</div>
                            <div class="col-md-8">{{ $client->user_id }}</div>
                        </div>

                        <div class="row mb-3">
                            <div class="col-md-4 font-weight-bold">{{ __(key: 'Name') }}</div>
                            <div class="col-md-8">{{ $client->name }}</div>
                        </div>

                        <div class="row mb-3">
                            <div class="col-md-4 font-weight-bold">{{ __(key: 'Email') }}</div>
                            <div class="col-md-8">{{ $client->email }}</div>
                        </div>

                        <div class="row mb-3">
                            <div class="col-md-4 font-weight-bold">{{ __(key: 'Phone Number') }}</div>
                            <div class="col-md-8">{{ $client->phone_number }}</div>
                        </div>

                        <div class="row mb-3">
                            <div class="col-md-4 font-weight-bold">{{ __(key: 'Balance') }}</div>
                            <div class="col-md-8">{{ $client->balance }}</div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

<!-- Información adicional de la dirección, si es necesario -->
@if($client->address_street || $client->city || $client->state || $client->postal_code || $client->country)
    <div class="row mb-3">
        <div class="col-md-4 font-weight-bold">{{ __(key: 'Address') }}</div>
        <div class="col-md-8">
            {{ $client->address_street }}, {{ $client->city }}<br>
            {{ $client->state }} {{ $client->postal_code }}<br>
            {{ $client->country }}
        </div>
    </div>
@endif

<!-- Aquí puedes incluir más campos según sea necesario -->

</div>

<div class="card-footer">
    <form action="{{ route(name: 'clients.destroy', parameters: $client->id) }}" method="POST">
        @csrf
        @method('DELETE')
        <button type="submit" class="btn btn-danger btn-sm" onclick="event.preventDefault(); confirm('Are you sure to delete this profile?') ? this.closest('form').submit() : false;">
            {{ __(key: 'Delete Profile') }}
        </button>
    </form>
</div>
</div>
</div>
</div>
@endsection

```

 Estructura de la vista **client.form**

```

@extends(view: 'layouts.app')

@section(section: 'template_title')
    {{ __(key: 'Client_Profile') }}
@endsection

@section(section: 'content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">
                        {{ __(key: 'Client_Profile') }}
                    </div>
                    <div class="card-body">
                        @if(session(key: 'error'))
                            <div class="alert alert-danger">
                                {{ session(key: 'error') }}
                            </div>
                        @endif

                        <!-- Formulario de cliente -->
                        <form action="{{ route(name: 'clients.update', parameters: $client->id) }}" method="POST">
                            @csrf
                            @method('PUT')

                            <!-- Campo de User ID -->
                            <div class="form-group mb-3">
                                <label for="user_id" class="form-label">{{ __(key: 'User_Id') }}</label>
                                <input type="text" class="form-control @error('user_id') is-invalid @enderror" name="user_id" id="user_id" value="{{ old(key: 'user_id', default: $client->user_id) }}" readonly>
                                @error('user_id')
                                    <div class="invalid-feedback">{{ $message }}</div>
                                @enderror
                            </div>

                            <!-- Campo de Nombre -->
                            <div class="form-group mb-3">
                                <label for="name" class="form-label">{{ __(key: 'Name') }}</label>
                                <input type="text" class="form-control @error('name') is-invalid @enderror" name="name" id="name" value="{{ old(key: 'name', default: $client->name) }}">
                                @error('name')
                                    <div class="invalid-feedback">{{ $message }}</div>
                                @enderror
                            </div>

                            <!-- Campo de Email -->
                            <div class="form-group mb-3">
                                <label for="email" class="form-label">{{ __(key: 'Email') }}</label>
                                <input type="email" class="form-control @error('email') is-invalid @enderror" name="email" id="email" value="{{ old(key: 'email', default: $client->email) }}">
                                @error('email')

```

```

                                    <div class="invalid-feedback">{{ $message }}</div>
                                @enderror
                            </div>

                            <!-- Campo de Número de Teléfono -->
                            <div class="form-group mb-3">
                                <label for="phone_number" class="form-label">{{ __(key: 'Phone_Number') }}</label>
                                <input type="text" class="form-control @error('phone_number') is-invalid @enderror" name="phone_number" id="phone_number" value="{{ old(key: 'phone_number', default: $client->phone_number) }}">
                                @error('phone_number')
                                    <div class="invalid-feedback">{{ $message }}</div>
                                @enderror
                            </div>

                            <!-- Campo de Balance -->
                            <div class="form-group mb-3">
                                <label for="balance" class="form-label">{{ __(key: 'Balance') }}</label>
                                <input type="number" class="form-control @error('balance') is-invalid @enderror" name="balance" id="balance" value="{{ old(key: 'balance', default: $client->balance) }}">
                                @error('balance')
                                    <div class="invalid-feedback">{{ $message }}</div>
                                @enderror
                            </div>

                            <!-- Botón de Submit -->
                            <div class="form-group mb-3">
                                <button type="submit" class="btn btn-primary">{{ __(key: 'Update_Profile') }}</button>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection

```

## 12 CSS

Css enlazado en app.blade.php para qué se apliquen los estilos a todas las vistas que extiendan del layout de app.blade.php.

```

/* Estilos Generales */
body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background-color: #f1f1f1; /* Fondo gris claro */
  background-image: url(public\selfinance_logo.jpeg);
  color: #333;
}

/* Estilos para la tarjeta principal */
.card {
  border-radius: 15px;
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);
  background-color: #ffffff;
  border: none;
}

/* Cabecera de la tarjeta */
.card-header {
  background-color: #28a745; /* Verde */
  color: white;
  text-align: center;
  padding: 20px;
  border-top-left-radius: 15px;
  border-top-right-radius: 15px;
}

.card-header h3 {
  font-size: 2rem;
  letter-spacing: 1px;
  font-weight: 600;
  margin: 0;
}

/* Contenido de la tarjeta */
.card-body {
  background-color: #f8f9fa; /* Gris suave */
  padding: 30px;
  border-radius: 10px;
}

/* Estilo para el mensaje de éxito */
.alert-success {
  background-color: #28a745; /* Verde */
  color: white;
  border-radius: 8px;
  padding: 10px;
  margin-bottom: 20px;
}

```

```

.alert-success strong {
  font-weight: bold;
}

/* Botón de acción */
.btn {
  border-radius: 50px;
  background-color: #28a745; /* Verde */
  color: white;
  border: none;
  box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
  transition: all 0.3s ease;
  padding: 10px 20px;
}

.btn:hover {
  background-color: #218838; /* Verde más oscuro al pasar el ratón */
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);
}

/* Títulos y texto en el cuerpo */
h5 {
  font-size: 1.4rem;
  font-weight: 400;
  color: #333;
}

p.lead {
  font-size: 1.3rem;
  font-weight: 300;
  color: #555;
}

/* Fondo y espaciado de los contenedores */
.container {
  max-width: 100%;
  margin-top: 4rem;
  margin-bottom: 4rem;
}

.row.justify-content-center {
  margin-top: 2rem;
}

.col-md-8 {
  margin: 0 auto;
}

```

```
/* Estilo de texto y alineación */
.text-center {
    text-align: center;
}

/* Alertas en forma de "toast" */
.alert-dismissible {
    border-radius: 8px;
}

.close {
    color: white;
    font-size: 1.2rem;
    opacity: 0.7;
    text-shadow: 0 0 5px rgba(0,0,0,0.5);
}

.close:hover {
    opacity: 1;
}

/* Estilos de sombra en las tarjetas */
.shadow-lg {
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2);
}

.my-5 {
    margin-top: 4rem;
    margin-bottom: 4rem;
}

/* Márgenes alrededor de los elementos */
.mb-5 {
    margin-bottom: 2rem;
}
```

```
/* Diseño Responsivo */
@media (max-width: 768px) {
  .card-body {
    padding: 20px;
  }

  .card-header h3 {
    font-size: 1.8rem;
  }

  .btn {
    width: 100%;
    padding: 12px;
    font-size: 1.1rem;
  }
}
```

## 13 Mejoras/Ampliaciones

En cuanto a mejoras o ampliaciones están:

- **Implementación de un Sistema de Roles**

Con distintos niveles de permisos.

- **Gráficos e Informes Financieros**

Implementar dashboards con estadísticas y gráficos interactivos sobre ingresos, gastos y balances y generación de reportes descargables en PDF o Excel.

- **Integración de Criptomonedas**

Soporte para visualizar y gestionar inversiones en criptomonedas con actualización en tiempo real.


- **Automatización de Transacciones**

Creación de reglas para automatizar pagos recurrentes y categorización de gastos e integración con APIs bancarias para actualizar automáticamente los movimientos financieros.

## **Soporte**

 Email: [devsupport@selfinance.com](mailto:devsupport@selfinance.com)

 Web: [www.selfinance.com](http://www.selfinance.com)

 Teléfono: +34 123 456 789