

Curso: SEPE09-23: Desarrollo web BackEnd

Docente: David Alcolea

## PLA 11\_ DETALLE Y DESCRIPCIÓN

**Nombre del PLA:** LARAVEL: Introducción, gestor de plantillas Blade y migraciones

### Objetivos del PLA:

- Construir una aplicación siguiendo el patrón Modelo Vista Controlador
- Introducción al framework Laravel 8: instalación y creación de un proyecto
- Uso de plantillas con Blade
- Creación de bases de datos utilizando migraciones
- Pruebas unitarias con PHPUnit
- Seeders

### Competencias asociadas al PLA:

| Competencias técnicas   | Soft Skills  |
|---|--|
| <ul style="list-style-type: none"><li>• Construcción de aplicaciones MVC</li><li>• Aprender el funcionamiento del framework Laravel</li><li>• Confección de plantillas con Blade</li><li>• Migraciones en laravel</li><li>• Pruebas unitarias</li></ul> | <ul style="list-style-type: none"><li>• Resolución de problemas</li><li>• Interpretar requerimientos</li><li>• Gestionar un proyecto</li><li>• Búsqueda, gestión y uso de la información</li></ul> |

### Calendario y horas de dedicación:

Del 22/06/23 al 28/06/23

|              |                       |    |
|--------------|-----------------------|----|
| Síncronas    | Webinars              | 4  |
|              | Tutorías presenciales | 0  |
| Asíncronas   |                       | 16 |
| Presenciales |                       |    |
| Totales      |                       | 20 |

**Modalidad PLA:** Ejercicio por etapas

### Instrucciones que recibirá el alumno para resolver el reto:

A partir de las vistas que proporcionará el profesor, el alumno creará un proyecto con el Framework Laravel 8 para, en esta primera actividad, integrarlas dentro del mismo utilizando el gestor de plantillas **Blade**, activar el enrutado entre las vistas y confeccionar la base de datos a utilizar.

En sucesivas actividades se irá completando el proyecto hasta construir una plataforma de mantenimiento de un producto asociado a la operativa de pagos con tarjeta de una Entidad financiera real (resumida por motivos obvios de duración del curso)

En este documento se describe, en el apartado ‘Requerimientos de usuario’ el funcionamiento de toda la plataforma al completo. En este PLA únicamente hay que realizar las siguientes actividades:

- Confección de las plantillas **Blade** de las vistas
- Confección del fichero de rutas para poder enlazar las vistas
- Confección del fichero de pruebas unitarias con phpunit
- Confección de los ficheros de migración de la base de datos a utilizar

### Instrucciones metodológicas para la entrega de los retos:

En cada actividad se especificarán los ficheros que se tendrán que subir a la plataforma Moodle del curso, y que serán diferentes en cada caso.

IMPORTANTE: No se debe subir a la plataforma ningún otro fichero o carpeta distinto a los indicados por el profesor para evitar generar un zip demasiado pesado.

En esta primera entrega de la plataforma el alumno deberá subir las siguientes carpetas del framework:

- carpeta **resources/views**
- archivo **routes/web.php**
- carpeta **app/Http/Controllers**
- carpeta **tests/Feature**
- carpeta **database/migrations**

No se debe subir ningún otro archivo o carpeta que no sean los anteriormente especificados

Se recomienda no modificar los archivos de las carpetas **css** y **img**, que se adjuntan como recursos del PLA, para evitar problemas con los ficheros que utilizará el formador para la corrección del ejercicio

El formato de entrega de la actividad es el siguiente:

**Pla14\_tu\_nombre\_tu\_apellido.zip**

La entrega del fichero será a través de la aplicación Moodle del curso.

Las actividades son individuales y intransferibles.

La resolución de dudas o comentarios se podrán realizar a través de las herramientas de comunicación de la aplicación Moodle del curso

La actividad se entregará antes de la fecha límite de entrega.

## Requerimientos de usuario (PLA 11: LARAVEL: Introducción, Blade y migraciones)

Se necesita crear una plataforma para realizar la operativa de alta y mantenimiento de un nuevo producto bancario denominado '**Cuenta Puntos**' para la entidad **ChungoBank Investments and Trusts**, en adelante **ChungoBank**, consistente en la acumulación de puntos, canjeables por obsequios de un catalogo, por la utilización de las tarjetas de crédito por parte de los clientes de la entidad.

### Entrada a la plataforma

La entrada a la plataforma se realizará desde la pantalla de **Gestión comercial**, en donde informaremos el nif de la persona a asociar al programa cuenta puntos o bien, en caso que la persona ya tenga una cuenta, realizar la consulta y mantenimiento de la misma

### Opciones de menú disponibles

Gestión Comercial → Pantalla de entrada a la plataforma (mostrada arriba)

Cuenta Puntos → Pantalla de alta, consulta y mantenimiento de la cuenta Puntos

Alta Personas → Pantalla de alta de personas

### Datos de la pantalla

*Identificación fiscal* → Obligatorio. Se informará con el nif de la persona a consultar

*Nombre y apellidos* → Protegido. Se mostrará nombre y apellidos de la persona consultada

### Acciones posibles en la pantalla

- Al completar el campo de identificación fiscal se realizará la consulta del nombre de la persona al abandonar el cursor el control del formulario.
- El nombre se mostrará en el control protegido situado a su derecha

### **Pantalla de alta de personas**

Pantalla para efectuar el alta de personas a asociar al programa de puntos

The screenshot shows a web interface for 'Chungobank Investments & Trusts'. At the top, there is a navigation bar with 'Gestión comercial' and 'Cuenta Puntos | Alta personas'. The main title is 'Alta Personas'. The form contains the following fields: NIF (single line), NOMBRE (single line), APELLIDOS (single line), DIRECCION (single line), EMAIL (single line), and TARJETA (four separate boxes for each digit). At the bottom, there are two buttons: 'Alta' and 'Abandonar', followed by the text 'Zona de mensajes'.

### Acceso a la pantalla

Se accederá a partir de la opción de menú 'Alta personas'

### Datos de la pantalla

*Nif* → Obligatorio. Se informará con el nif de la persona a dar de alta

*Nombre* → Obligatorio. Se informará con el nombre de la persona

*Apellidos* → Obligatorio. Se informará con los apellidos de la persona

*Dirección* → Obligatorio. Se informará con la dirección de la persona

*Email* → Opcional. Se informará con el email de la persona

*Tarjeta* → Protegido. Se mostrará el número PAN que el sistema asignará a la tarjeta

### Acciones posibles en la pantalla

#### **Botón Alta**

- Estado por defecto: Activo
- Función:

Al completar todos los controles del formulario y pulsar el botón se procederá al alta de la persona en la base de datos (tabla Personas)

- Desactivación: Nunca

### Botón Abandonar

- Estado por defecto: Activo
- Función: Retorno a la pantalla de *Gestión comercial*
- Desactivación: Nunca

### Pantalla de alta y mantenimiento cuenta puntos

Pantalla para efectuar el alta, consulta y mantenimiento de la cuenta puntos de la persona consultada en la pantalla de Gestión Comercial

### Acceso a la pantalla

Se accederá a partir de la opción de menú 'Cuenta Puntos'

### Datos de la pantalla

- |                           |  |
|---------------------------|--|
| <i>Contrato puntos</i>    | → Protegido. Se mostrará entidad, oficina, dc y numero de cuenta puntos de la persona consultada (si ya tiene una) o el numero asignado por el sistema en caso de alta |
| <i>Titular</i>            | → Protegido. Se mostrará nombre y apellidos de la persona seleccionada   |
| <i>Código programa</i>    | → Obligatorio. Mostrar/seleccionar el programa de puntos   |
| <i>Descripción</i>        | → Protegido. Se mostrará la descripción del programa seleccionado  |
| <i>Renuncia extracto</i>  | → Opcional. Se seleccionará si el cliente no desea recibir extracto mensual  |
| <i>Renuncia obtención</i> | → Opcional. Se seleccionará si el cliente no desea acumular puntos   |

### Acciones posibles en la pantalla

#### Botón Alta

- Estado por defecto:
  - Activo si el cliente seleccionado en Gestión comercial carece de cuenta puntos
  - Inactivo si el cliente seleccionado en Gestión comercial tiene una cuenta puntos

- **Función:**  
Al completar todos los controles del formulario y pulsar el botón se procederá al alta de la cuenta en la base de datos (tabla Cuentas)
- **Activación:** Al dar de baja una cuenta
- **Desactivación:** Al completar el alta de la cuenta

#### *Botón Modificar*

- **Estado por defecto:**  
Activo si el cliente seleccionado en Gestión comercial tiene una cuenta puntos  
Inactivo si el cliente seleccionado en Gestión comercial carece de cuenta puntos
- **Función:**  
Una vez realizada la consulta de la cuenta de la persona seleccionada en Gestión Comercial, al completar todos los controles del formulario y pulsar el botón se procederá a la modificación de la cuenta en la base de datos (tabla Cuentas)
- **Activación:** Al dar de alta una cuenta
- **Desactivación:** Al dar de baja una cuenta

#### *Botón Baja*

- **Estado por defecto:**  
Activo si el cliente seleccionado en Gestión comercial tiene una cuenta puntos  
Inactivo si el cliente seleccionado en Gestión comercial carece de cuenta puntos
- **Función:**  
Una vez realizada la consulta de la cuenta de la persona seleccionada en Gestión Comercial, y pulsar el botón se procederá a la baja de la cuenta en la base de datos (tabla Cuentas)
- **Activación:** Al dar de alta una cuenta
- **Desactivación:** Al dar de baja una cuenta

#### *Botón Consulta Mvtos*

- **Estado por defecto:**  
Activo si el cliente seleccionado en Gestión comercial tiene una cuenta puntos  
Inactivo si el cliente seleccionado en Gestión comercial carece de cuenta puntos
- **Función:**  
Una vez realizada la consulta de la cuenta de la persona seleccionada en Gestión Comercial, y pulsar el botón se enlazará con la pantalla de consulta de movimientos de la cuenta
- **Activación:** Al dar de alta una cuenta
- **Desactivación:** Al dar de baja una cuenta

### Botón Abandonar

- Estado por defecto: Activo
- Función: Retorno a la pantalla de Gestión comercial
- Desactivación: Nunca

## Pantalla de consulta movimientos cuenta puntos

Pantalla para efectuar la consulta de los movimientos de una cuenta puntos utilizando filtros

**Chungobank Investments & Trusts**  
Gestión comercial Cuenta Puntos Alta personas

**Consulta movimientos Cta Puntos**

CONTRATO PUNTOS: [ ] [ ] [ ] [ ]

FECHA DESDE: dd/mm/aaaa [ ] FECHA HASTA: dd/mm/aaaa [ ]

OPERACIÓN:  
☐ Asignación  
☐ Disposición  
☒ Todos

CONCEPTO [ ]

Últimos ☐

ULTIMO EXTRACTO: dd/mm/aaaa SALDO PUNTOS: [ ]

|                       | FECHA      | A/D | CONCEPTO  | PUNTOS | SALDO  |
|-----------------------|------------|-----|---|--------|--------|
| <input type="radio"/> | 30/05/2019 | A   | Ejemplo de información de detalle de movimiento | 500    | 25.000 |

Aceptar Detalle Imprimir Abandonar Alta movimiento Zona de mensajes

### Acceso a la pantalla

Se accederá a partir de la pantalla 'Alta y mto cuenta puntos'

### Datos de la pantalla

|                 |   |
|-----------------|---|
| Contrato puntos | → Protegido. Se mostrará entidad, oficina, dc y numero de cuenta puntos   |
| Fecha desde     | → Opcional. Fecha a partir de la cual se consultarán los movimientos      |
| Fecha hasta     | → Opcional. Fecha final de consulta de los movimientos                    |
| Operación       | → Obligatorio, marcado 'Todos'. Tipo de movimiento (suma o resta puntos)  |
| Concepto        | → Opcional. Filtro de movimientos por el concepto introducido             |
| Últimos         | → Opcional, marcado por defecto. Mostrar solo los últimos 10 movimientos  |
| Último extracto | → Protegido. Se mostrará la fecha de envío del último extracto al cliente |
| Saldo puntos    | → Protegido. Saldo total de puntos en el momento de la consulta           |

#### Listado de movimientos:

|           |   |
|-----------|---|
| Selección | → Obligatorio. Selección del movimiento a consultar en detalle              |
| Fecha     | → Fecha del movimiento de puntos  |
| A/D       | → Indicador de asignación (acumular puntos) o disposición (gasto de puntos) |
| Concepto  | → Concepto identificativo del movimiento                                    |

**Puntos** → Puntos acumulados o consumidos en el movimiento

**Saldo** → Saldo de puntos después del movimiento

### Acciones posibles en la pantalla

#### *Botón Aceptar*

- Estado por defecto: Inactivo
- Función:
  - Realiza la consulta de movimientos a partir de los criterios de búsqueda especificados
- Activación: Al modificar un criterio de búsqueda
- Desactivación: Al completar la consulta de movimientos

#### *Botón Detalle*

- Estado por defecto: Inactivo
- Función:
  - Una vez seleccionado un movimiento de la lista enlaza con la pantalla de detalle de movimiento
- Activación: Al seleccionar un movimiento de la lista
- Desactivación: Nunca

#### *Botón Imprimir*

- Estado por defecto:
  - Activo si la consulta inicial devuelve movimientos
  - Inactivo si la consulta inicial no devuelve movimientos
- Función:
  - Una vez realizada la consulta imprime la lista de movimientos de la cuenta puntos

|                                  | FECHA      | A/D | CONCEPTO  | PUNTOS | SALDO  |
|----------------------------------|------------|-----|---|--------|--------|
| <input type="radio"/>            | 30/05/2019 | A   | Ejemplo de información de detalle de movimiento | 500    | 25.000 |
| <input checked="" type="radio"/> | 2019-07-23 | A   | concepto 1                                      | 250    | 250    |
| <input type="radio"/>            | 2019-07-30 | D   | concepto 1                                      | 250    | 0      |
| <input type="radio"/>            | 2019-07-23 | A   | concepto 3                                      | 500    | 500    |

- Activación: La consulta devuelve movimientos
- Desactivación: La consulta no devuelve movimientos

#### *Botón Abandonar*

- Estado por defecto: Activo
- Función: Retorno a la pantalla de Alta y mantenimiento cuenta puntos
- Desactivación: Nunca



### Botón Alta movimiento

- Estado por defecto: Activo
- Función: Acceso a la pantalla de Alta de movimiento cuenta puntos
- Desactivación: Nunca

### Pantalla de alta movimiento cuenta puntos

Pantalla para efectuar el alta manual de movimientos de una cuenta puntos

The screenshot shows a web application interface for 'Chungobank Investments & Trusts'. The main title is 'Alta movimientos Cta Puntos'. The form contains the following fields and options:

- CONTRATO PUNTOS: Four input boxes for contract details.
- FECHA MOVIMIENTO: A date picker showing 'dd/mm/aaaa'.
- OPERACIÓN: Two radio buttons, 'Asignación' (selected) and 'Disposición'.
- CONCEPTO: A single-line text input field.
- SALDO: A single-line text input field.
- PUNTOS: A single-line text input field.

At the bottom, there are two buttons: 'Alta' and 'Abandonar', followed by the text 'Zona de mensajes'.

### Acceso a la pantalla

Se accederá a partir de la pantalla 'Consulta movimientos cuenta puntos'

### Datos de la pantalla

- |                         |   |
|-------------------------|---|
| <i>Contrato puntos</i>  | → Protegido. Se mostrará entidad, oficina, dc y numero de cuenta puntos |
| <i>Fecha movimiento</i> | → Obligatorio, por defecto fecha del día. Fecha del movimiento          |
| <i>Operación</i>        | → Obligatorio, por defecto 'Asignación'. Tipo de movimiento             |
| <i>Concepto</i>         | → Obligatorio. Concepto a asociar al movimiento (libre entrada)         |
| <i>Saldo</i>            | → Protegido. Saldo total de puntos antes y después del alta             |
| <i>Puntos</i>           | → Obligatorio. Número de puntos a sumar o restar                        |

### Acciones posibles en la pantalla

#### Botón Alta

- Estado por defecto: Activo
- Función:  
Realiza el alta del movimiento en la base de datos a partir de la información introducida en el formulario
- Desactivación: Nunca

### Botón Abandonar

- Estado por defecto: Activo
- Función: Retorno a la pantalla de Consulta movimientos cuenta puntos
- Desactivación: Nunca

### Pantalla de detalle movimiento cuenta puntos

Pantalla para consultar el detalle de un movimiento seleccionado en la pantalla de Consulta Movimientos Cuenta Puntos

### Acceso a la pantalla

Se accederá a partir de la pantalla 'Consulta movimientos cuenta puntos'

### Datos de la pantalla

|                         |  |
|-------------------------|--|
| <i>Contrato puntos</i>  | → Protegido. Se mostrará entidad, oficina, dc y numero de cuenta puntos    |
| <i>Fecha movimiento</i> | → Protegido. Mostrará la fecha del movimiento                              |
| <i>Concepto</i>         | → Protegido. Concepto asociado al movimiento                               |
| <i>Tarjeta</i>          | → Protegido. Número PAN de tarjeta asociada al contrato de puntos          |
| <i>Localizador</i>      | → Protegido. Código localizador del comercio donde se realizó la operación |
| <i>Comercio</i>         | → Protegido. Nombre del comercio donde se realizó la operación             |
| <i>Comentarios</i>      | → Protegido. Comentarios añadidos al movimiento                            |

### Acciones posibles en la pantalla

### Botón Abandonar

- Estado por defecto: Activo
- Función: Retorno a la pantalla de Consulta movimientos cuenta puntos
- Desactivación: Nunca

## Especificaciones estructura plataforma

Toda la plataforma se realizará sobre el framework Laravel 8 utilizando todas las herramientas del mismo en cuanto a:

- Confección de las vistas utilizando el gestor de plantillas *Blade*
- Utilización de las directivas Laravel para mostrar información dinámica en las vistas
- Utilización de controladores para la operativa de negocio y la carga de las vistas
- Utilización de modelos para el acceso a la base de datos
- Confección de los accesos a la base de datos utilizando *Query Builder* o Eloquent ORM de Laravel (con preferencia para Eloquent)
- Creación de la base de datos utilizando el sistema de migraciones de Laravel
- Incorporación de datos a las tablas utilizando el sistema de *Seeders* de Laravel
- Confección de los ficheros de pruebas unitarias con phpUnit

## Explicación técnica del Reto

### EJERCICIO 1: CREACIÓN DEL PROYECTO

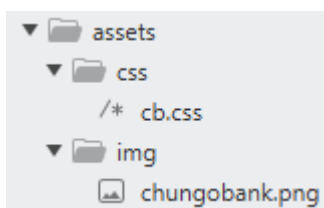
Crearemos un proyecto llamado **ChungoBank** utilizando Laravel 8

### EJERCICIO 2: CARGA DE LAS VISTAS CON BLADE

En la carpeta **resources/views** crearemos todas las plantillas **blade** de la plataforma a partir de los documentos **html** que entregará el profesor y que son los siguientes:

```
<> alta-movimientos.html  
<> alta-mto-puntos.html  
<> alta-personas.html  
<> consulta-movimientos.html  
<> detalle-movimiento.html  
<> gestion.html
```

Y los archivos **css** y **img** que también se proporcionan en la carpeta **assets**



NOTA: Se recomienda no modificar los estilos css

A partir de los documentos **html** proporcionados se deberán crear los componentes de las vistas para tener: un único documento con la estructura común de todas las páginas (**layout**) y seis componentes que corresponderán, cada uno de ellos, a las operativas a realizar en la plataforma.

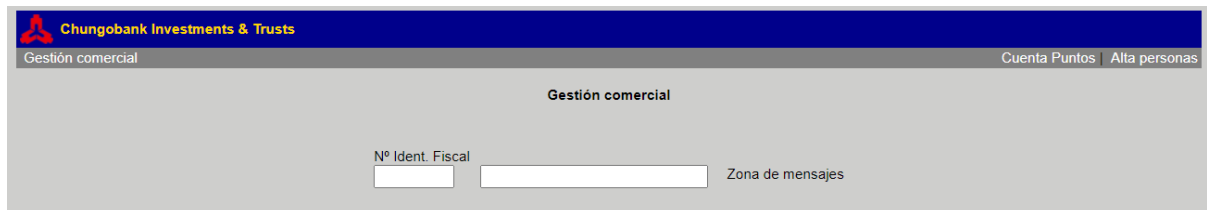
Se consideran estructuras comunes las siguientes

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Chungo Bank</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="assets/css/cb.css">  
  </head>  
  <body>  
    <div class='contenedor'>  
      <header>  
          
        <h4>Chungobank Investments & Trusts</h4>  
      </header>  
      <nav>  
        <div><a href="gestion.html">Gestión comercial</a></div>  
        <div><a href="alta-mto-puntos.html">Cuenta Puntos</a> |  
        <a href="alta-personas.html">Alta personas</a></div>  
      </nav>  
      <h4 class='center'>Gestión comercial</h4>  
      <section>==</section>  
    </div>  
  </body>  
</html>
```

Siendo, por tanto, el contenido de cada uno de los 6 componentes el que se encuentra dentro de la etiqueta `<section>`

### EJERCICIO 3: HABILITAR LAS RUTAS PARA LA NAVEGACION ENTRE PANTALLAS

Una vez tenemos las vistas creadas, al entrar en la plataforma, se mostrará la pantalla de Gestión comercial



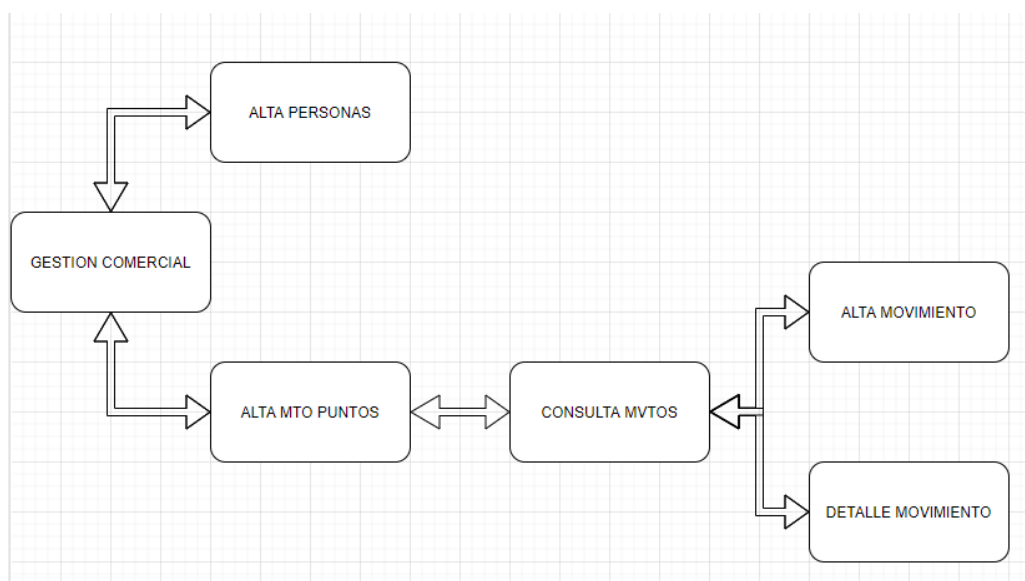
Para ello tendremos que modificar el fichero de rutas en **routes/web.php** para indicar que la raíz del proyecto sea nuestra vista en vez de la vista por defecto que trae Laravel (y que no la necesitamos)

Tendremos que crear también un controlador en **app/Http/Controllers** (por ejemplo **CargaVistasController**) para realizar la carga de las vistas desde el controlador y no en el fichero de rutas

A partir de aquí, habilitaremos la navegación entre todas las pantallas de la aplicación, ya sea utilizando las opciones del menú superior, como los botones habilitados en cada vista para enlazar con otras pantallas.

Tendremos que indicar las rutas a los controladores en el fichero **routes/web.php** y, además modificar los enlaces de la barra de menú y los botones para que apunten a estas rutas

El árbol de navegación es el siguiente:



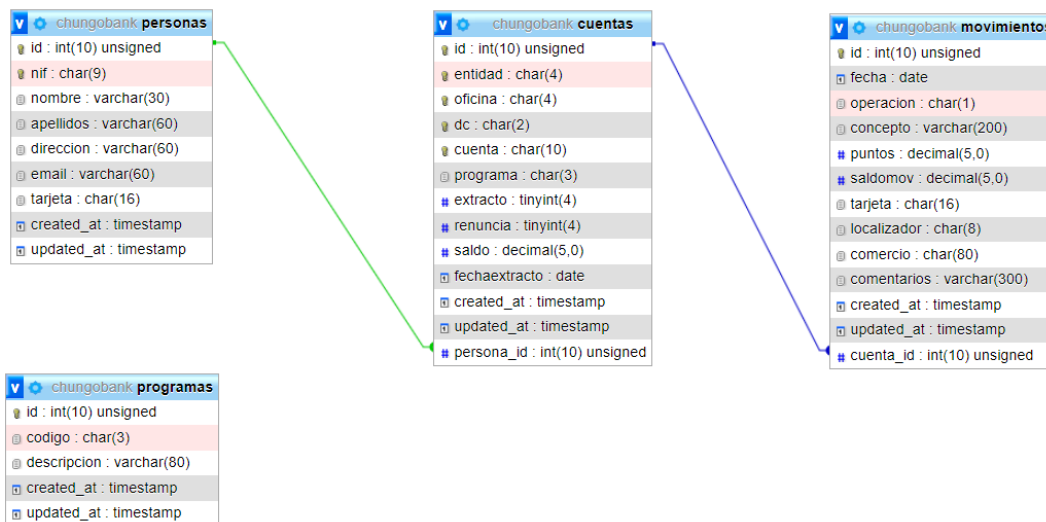
## NOTAS:

- En cada pantalla debe aparecer el título correcto que corresponda a cada operativa
- Los enlaces del menú apuntarán a las rutas correctas según se especifica en la documentación que encontrareis en recursos del PLA
- Para asociar una ruta a un botón de forma provisional, para poder probar la carga de vistas, utilizaremos el siguiente código. Ejemplo para el botón *Abandonar* de la pantalla 'Alta Persona':

```
<input type="button" id="salir" value='Abandonar' onclick="window.location.href = &#39;{{ url('/') }}&#39;">
```

## EJERCICIO 4: CONSTRUCCIÓN DE LA BASE DE DATOS

Vamos a elaborar la base de datos que utilizaremos en las diferentes actividades de la plataforma. Para ello vamos a utilizar el sistema de migraciones de Laravel de forma que podemos construirla sin necesidad de utilizar el panel de *phpMyAdmin* o *MySQL Workbench*



**PREGUNTA:** ¿Podéis detectar que Forma Normal está incumpliendo el diseño anterior y por cuál motivo?

El primer paso será crear la base de datos vacía de tablas (este paso no lo podemos realizar con el sistema de migraciones). Para ello en *phpMyAdmin* creamos la base de datos **chungobank**

Con la base de datos ya creada podemos utilizar el sistema de migraciones para la creación de las entidades, atributos y relaciones con la estructura que se especifica a continuación

Personas

| # | Nombre     | Tipo        | Cotejamiento       | Atributos | Nulo | Predeterminado |
|---|------------|-------------|--------------------|-----------|------|----------------|
| 1 | id         | int(10)     |                    | UNSIGNED  | No   | Ninguna        |
| 2 | nif        | char(9)     | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 3 | nombre     | varchar(30) | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 4 | apellidos  | varchar(60) | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 5 | direccion  | varchar(60) | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 6 | email      | varchar(60) | utf8mb4_unicode_ci |           | Si   | NULL           |
| 7 | tarjeta    | char(16)    | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 8 | created_at | timestamp   |                    |           | Si   | NULL           |
| 9 | updated_at | timestamp   |                    |           | Si   | NULL           |

```
INSERT INTO `personas` (`id`, `nif`, `nombre`,  
`apellidos`, `direccion`, `email`, `tarjeta`, `created_at`,  
`updated_at`) VALUES (NULL, '10000001A', 'Margaret',  
'Rose', 'Av. Pignarelli, 56', 'margaret@mail.com',  
'1234567890123456', CURRENT_TIMESTAMP,  
CURRENT_TIMESTAMP);
```

## Cuentas

| #  | Nombre        | Tipo         | Cotejamiento       | Atributos | Nulo | Predeterminado |
|----|---------------|--------------|--------------------|-----------|------|----------------|
| 1  | id 🗝️         | int(10)      |                    | UNSIGNED  | No   | Ninguna        |
| 2  | entidad 🗑️    | char(4)      | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 3  | oficina 🗑️    | char(4)      | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 4  | dc 🗑️         | char(2)      | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 5  | cuenta 🗑️     | char(10)     | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 6  | programa      | char(3)      | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 7  | extracto      | tinyint(4)   |                    |           | No   | Ninguna        |
| 8  | renuncia      | tinyint(4)   |                    |           | No   | Ninguna        |
| 9  | saldo         | decimal(5,0) |                    |           | No   | 0              |
| 10 | fechaextracto | date         |                    |           | Sí   | NULL           |
| 11 | created_at    | timestamp    |                    |           | Sí   | NULL           |
| 12 | updated_at    | timestamp    |                    |           | Sí   | NULL           |
| 13 | persona_id 🗑️ | int(10)      |                    | UNSIGNED  | No   | Ninguna        |

NOTA: Entidad, oficina, dc y cuenta es un único índice múltiple

## Movimientos

| #  | Nombre       | Tipo         | Cotejamiento       | Atributos | Nulo | Predeterminado      |
|----|--------------|--------------|--------------------|-----------|------|---------------------|
| 1  | id 🗝️        | int(10)      |                    | UNSIGNED  | No   | Ninguna             |
| 2  | fecha        | date         |                    |           | No   | Ninguna             |
| 3  | operacion    | char(1)      | utf8mb4_unicode_ci |           | No   | Ninguna             |
| 4  | concepto     | varchar(200) | utf8mb4_unicode_ci |           | No   | Ninguna             |
| 5  | puntos       | decimal(5,0) |                    |           | No   | Ninguna             |
| 6  | saldomov     | decimal(5,0) |                    |           | No   | Ninguna             |
| 7  | tarjeta      | char(16)     | utf8mb4_unicode_ci |           | No   | Ninguna             |
| 8  | localizador  | char(8)      | utf8mb4_unicode_ci |           | Sí   | NULL                |
| 9  | comercio     | char(80)     | utf8mb4_unicode_ci |           | Sí   | NULL                |
| 10 | comentarios  | varchar(300) | utf8mb4_unicode_ci |           | Sí   | NULL                |
| 11 | created_at   | timestamp    |                    |           | No   | current_timestamp() |
| 12 | updated_at   | timestamp    |                    |           | Sí   | NULL                |
| 13 | cuenta_id 🗑️ | int(10)      |                    | UNSIGNED  | No   | Ninguna             |

## Programas

| # | Nombre      | Tipo        | Cotejamiento       | Atributos | Nulo | Predeterminado |
|---|-------------|-------------|--------------------|-----------|------|----------------|
| 1 | id 🗝️       | int(10)     |                    | UNSIGNED  | No   | Ninguna        |
| 2 | codigo 🗑️   | char(3)     | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 3 | descripcion | varchar(60) | utf8mb4_unicode_ci |           | No   | Ninguna        |
| 4 | created_at  | timestamp   |                    |           | Sí   | NULL           |
| 5 | updated_at  | timestamp   |                    |           | Sí   | NULL           |

NOTA IMPORTANTE: Tened cuidado que los nombres de las tablas, los atributos y el tipo de dato de cada uno de ellos coincidan exactamente con los mostrados en las estructuras que se muestran arriba

```
INSERT INTO `cuentas` (`id`, `entidad`, `oficina`, `dc`,  
`cuenta`, `programa`, `extracto`, `renuncia`, `saldo`,  
`fechaextracto`, `created_at`, `updated_at`,  
`persona_id`) VALUES (NULL, '0001', '0200', '10',  
'0200123456', 'PPP', '0', '0', '0', '2021-11-16',  
CURRENT_TIMESTAMP, CURRENT_TIMESTAMP, '55');
```

```
INSERT INTO `movimientos` (`id`, `fecha`, `operacion`,  
`concepto`, `puntos`, `saldomov`, `tarjeta`,  
`localizador`, `comercio`, `comentarios`, `created_at`,  
`updated_at`, `cuenta_id`) VALUES (NULL, '2021-11-  
16', 'A', 'Compra en Comercio On-Line', '100', '100',  
'1234567890123456', 'AA233EE', 'El Libro Feliz',  
'Pendiente de verificación', CURRENT_TIMESTAMP,  
CURRENT_TIMESTAMP, '8');
```

```
INSERT INTO `programas` (`id`, `codigo`, `descripcion`,  
`created_at`, `updated_at`) VALUES (NULL, 'PPP',  
'Programa Puntos de Prueba', CURRENT_TIMESTAMP,  
CURRENT_TIMESTAMP);
```

## EJERCICIO 5: ELABORACIÓN DEL FICHERO DE PRUEBAS UNITARIAS

Como último paso de esta actividad, vamos a elaborar el fichero de pruebas unitarias en la carpeta **tests/Feature** utilizando PHPUnit con el objetivo de probar toda la carga de vistas.

El fichero de pruebas podría llamarse **CargaVistasTest** y, al ejecutarlo con **phpUnit**, debería aparecer un mínimo de 7 test con 13 aserciones

```
C:\xampp\htdocs\CIFO\PLA14\RESUELTO\chungobank>vendor\bin\phpunit
PHPUnit 9.5.10 by Sebastian Bergmann and contributors.

.....                                     7 / 7 (100%)

Time: 00:00.245, Memory: 20.00 MB

OK (7 tests, 13 assertions)
```

NOTA: Hay que probar, como mínimo, que se carga correctamente la vista y que el título en cada una de ellas es el correcto

## EJERCICIO 6 (OPCIONAL): USO DE SEEDERS PARA CREAR DATOS

Como actividad adicional no evaluable, se propone que, una vez tengáis creadas las tablas de la plataforma, inserteis algunos datos de prueba utilizando los seeders de Laravel junto con la clase **faker**. Ejemplo para la tabla personas:

En **database/factories** creamos un archivo **PersonasFactory.php** con la definición de datos:

```
public function definition()
{
    return [
        'nif' => $this->faker->unique()->regexify('[A-Za-z0-9]{9}'),
        'nombre' => $this->faker->firstName(),
        'apellidos' => $this->faker->lastName(),
        'direccion' => $this->faker->text(30),
        'email' => $this->faker->unique()->safeEmail(),
        'tarjeta' => $this->faker->regexify('[0-9]{16}')
    ];
}
```

En **database/seeders** creamos el fichero **PersonasSeeder.php** para insertar 10 filas aleatorias:

```
public function run()
{
    Personas::factory(10)->create();
}
```





**Generalitat  
de Catalunya**



**Unió Europea**  
**Fons Social Europeu**  
L'FSE inverteix en el teu futur

### Recursos de aprendizaje vinculados:

Toda la documentación (manuales, vídeo tutoriales, esquemas, etc) se encontrarán en el aula virtual así como los recursos necesarios para resolver el reto (imágenes, etc)

- Laravel 01: Instalación y estructura
- Laravel 02: Vistas y layouts con Blade
- Laravel 03: Migraciones
- Laravel 04: Seeders