

Aplicación bancaria

Trabajo colaborativo de DES y DIW MARZO 2018

Andrei García Cuadra

Andrei García Cuadra

1.	Preparación y puesta en marcha	2
1.1.	Requerimientos	2
1.2.	Jerarquía, organización y orden	2
1.3.	Repositorio	2
1.4.	Rutas del api rest	3
2.	aplicación	4
2.1.	AJAX	4
2.2.	SASS	4
2.3.	Tratamiento de imágenes	4
2.4.	Accesibilidad	4
2.5.	Responsive	5
2.6.	Cross-browser	5
2.7.	Seguridad	5
2.8.	Autenticación	5
2.9.	Comprobación de formatos	5
2.9.	Menú y páginas	6
3.	Pruebas	7
3.1.	Pruebas de responsive	7
3.2.	Pruebas de html5	7
3.3.	Pruebas de accesibilidad	8
3.4.	Pruebas de css	8
4.	Base de datos	9
5.	SEO	9
6.	Aspecto y páginas de la aplicación	9
6.1.	Sitemap	9
6.2.	Galería de imágenes	10
7.	Anexos	13
7.1.	Aplicación externa (no oficial, con acceso api)	13
7.2.	Conexión de clientes	13
8.	Copyright y copyleft	13

Aplicación bancaria

1. PREPARACIÓN Y PUESTA EN MARCHA

La aplicación consta de 3 apartados (o subaplicaciones):

- **API REST:** Es de donde se recoge la información. Para poder usarlo se necesitan claves de la api. Está escrito en **Java 8**.
- **Aplicación bancaria (Vista):** Aplicación mediante la cual los clientes del banco podrán interactuar, ingresar dinero, retirarlo, abrir cuentas... Etcétera. Está escrito con **SASS 5, LESS**
- **Aplicación externa (Vista):** Aplicación de terceros que permite realizar ingresos en el banco.

1.1. REQUERIMIENTOS

El proyecto requiere de tener instaladas ciertas dependencias, en función de la aplicación a desarrollar.

- SASS ([Guía de instalación](#)).
- Servidor web (integrado).
- NPM. Proyecto e información



El proyecto ha sido desarrollado mediante el IDE Netbeans.

Además, integra la precompilación de SASS y optimizaciones a la hora de compilar el proyecto final, de modo que integra el entorno de desarrollo de producción y desarrollo en el IDE, y el entorno de producción en la compilación, facilitando así la rápida integración y despliegue de la aplicación.

1.2. JERARQUÍA, ORGANIZACIÓN Y ORDEN

Para entender este apartado, separaremos todo en el patrón de diseño MVC. Debemos distinguir dos tipos de programas:

- **Programas de servidor:** Su jerarquía estará ligada a M*C, de modo que obtendremos distintas capas relacionadas con el servidor, tales como la capa de DAO para el acceso a datos, la capa de SERVICIOS para el tratamiento de datos, la capa HTTP para el acceso mediante web (si es oportuno), etc. Llevando éstos nombres las carpetas (o paquetes) contenedores.
- **Programas de cliente:** Su jerarquía será de árbol, es decir, se crearán distintas carpetas para separar los tipos de archivos y en función de su rango de importancia, siendo la **carpeta base la raíz**, la **carpeta de estilos assets**, la **carpeta de imágenes img**, etc.

1.3. REPOSITORIO

El proyecto contiene un repositorio ubicado en bitbucket, es público y accesible por quien quiera. Además, se permite su lectura y libre distribución (copyleft).

<https://bitbucket.org/andi2garcia/app-banco>



1.4. RUTAS DEL API REST

Nuestro servidor de datos posee rutas de acceso siguiendo el estándar REST, y haciendo uso de los métodos OPTIONS, GET, PUT, POST, DELETE. Las rutas de acceso a datos son las siguientes:

- **/clientes**
 - GET
 - Devuelve los datos de una cuenta.
- **/cuenta/{numero_cuenta}**
 - GET
 - Devuelve información de la cuenta y de sus titulares.
 - PUT
 - Añade una cuenta y a sus titulares.
 - DELETE
 - Elimina una cuenta y sus movimientos
- **/login/{token}**
 - GET
 - Comprueba si el usuario está conectado mediante el token.
 - POST
 - Revisa las credenciales de conexión de un usuario y devuelve el token.
- **/movimientos**
 - GET
 - Devuelve el listado de movimientos global.
 - POST
 - Devuelve el listado de movimientos filtrados por fecha/cuenta.
- **/operaciones**
 - POST
 - Realiza un reintegro
 - PUT
 - Realiza un ingreso
- **/usuarios/{dni}**
 - GET
 - Devuelve el listado de todos los usuarios.
 - POST
 - Actualizar usuario.
 - PUT
 - Añadir un usuario.
 - DELETE
 - Borrar un usuario dado el dni del path param.



REST API

2. APLICACIÓN

La idea básica de esta aplicación es que tenga los colores y el estilo del banco BBVA. Asimismo, se ha utilizado el framework [Materialize](#) para otorgar un estilo acorde a lo requerido. El proyecto contiene varios archivos en la ruta base que pueden no ser de importancia si tan sólo se quiere la compilación. La compilación se ubica en `public_html`, y contiene los ficheros `*.html`, `*.css` y `*.js` ya compilados y minimizados por los precompiladores. La ruta base contiene los archivos para la compilación (NPM, SASS, LESS, ...). Se ha elegido minificar todos estos archivos para ahorrar ancho de banda. Se han cumplido con los estándares de HTML5, CSS3 y JavaScript ECDMA 6.



2.1. AJAX

Se ha utilizado para toda la aplicación peticiones asíncronas de AJAX (jQuery), para recibir todos los datos necesarios del api rest. Estas llamadas se han securizado por Origin y Api keys, como se comentará posteriormente. Los datos recibidos y enviados han sido siempre en el estándar JSON. Se han hecho uso de los métodos GET, PUT, POST y DELETE en la variable type de ajax, para poder enviar la necesaria petición al REST.

2.2. SASS

El fichero que contiene las variables que se han utilizado está ubicado en `sass/components/_variables.scss`. Se han modificado las variables nativas de materialize para cambiar la fuente de letras, el tamaño, los colores principales, varios aspectos del responsive, solución a errores internos (relacionados con las modales), etc.

Pueden verse todas las variables afectadas en el fichero anteriormente mencionado, además de los mixins.

Para poder apreciar un buen ejemplo de mixins, podríamos consultar el archivo `nexo` del proyecto `materialize.scss`.

2.3. TRATAMIENTO DE IMÁGENES

Como logotipo se ha utilizado el oficial del banco BBVA, el cual ha sido minimizado y comprimido con la herramienta photoshop. El resto de imágenes han sido tratadas para acordarlas al contenido de la página y, sobretodo, reducir su considerable tamaño (sobre todo en las imágenes de fondo). Todas las imágenes poseen copyrights relacionados con el banco BBVA, por lo tanto, son imágenes internas de la empresa, por lo tanto están en la más absoluta legalidad.



Para mantener la identidad de la página, se ha añadido un favicon con el mismo logo que el de la empresa.

2.4. ACCESIBILIDAD

Se ha presupuesto que las personas que utilizarán la aplicación tendrán una media de 25 a 40 años, en las cuales, la vista no es aún demasiado deficiente y los empleados además están algo formados.

El panel es simple y funcional, se busca la agilidad laboral y la rapidez en las operaciones.

2.5. RESPONSIVE

Se han utilizado siempre medidas relativas, además de un grid de 12 celdas optimizado para pantallas largas, medianas y pequeñas. De este modo, la web se visualizará correctamente desde cualquier dispositivo.

2.6. CROSS-BROWSER

El rest incluye la política de privacidad permisiva a las políticas de CORS. Esto se refiere a que el rest puede ser accedido desde jQuery y clientes web similares basados en JavaScript (Angular, HttpClient, etc).

2.7. SEGURIDAD

El rest incluye dos métodos de autenticación:

- **Autenticación por origen:** Se revisa la URL del cliente de la api, y si coincide con la oficial del banco, permite la consulta.
- **Autenticación por api key:** Si no se logra autenticar por origen, se revisa el header rest-api-key el cual ha de contener la clave api autorizada para el uso de la aplicación.

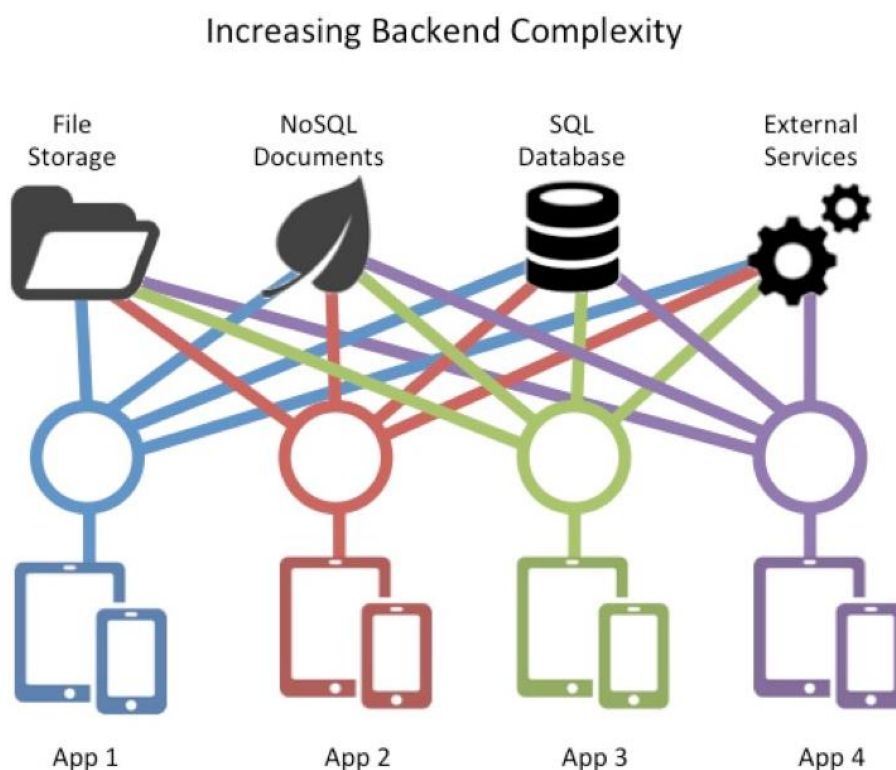
2.8. AUTENTICACIÓN

La autenticación de este cliente se logra utilizando JWT's (JSON Web Token), los cuáles son almacenados en localStorage

2.9. COMPROBACIÓN DE FORMATOS

Para comprobar los formatos de DNI y número de cuenta se han utilizado tanto los patrones de HTML5 como funciones de JavaScript.

La comprobación se ha realizado tanto de lado de servidor como de lado de cliente.



Aplicación bancaria

2.9. MENÚ Y PÁGINAS

El menú se ha tratado de simplificar al máximo, separando tan sólo las opciones básicas. Así, se ha llegado a tener las páginas básicas (excepto conexión) en el menú.

- Conexión.
 - Al conectar se comprueba que el PIN sea de 3 dígitos y el DNI sea válido tanto en el cliente como en el servidor.
 - Envía una petición AJAX al servidor sin api-key, ya que la cabecera el dominio es la misma y el servidor lo acepta.
- Ingresos y reintegros.
 - Comprueba los fondos de la cuenta.
 - Comprueba el formato de la cuenta.
 - Comprueba si las cuentas existen.
- Apertura de cuentas.
 - Comprueba que la cuenta no exista.
 - Comprueba la validez de los titulares.
 - Comprueba el formato de la cuenta.
 - Comprueba el formato del dni.
- Listado de movimientos.
 - Permite ver movimientos globales.
 - Permite ver movimientos entre fechas.
 - Permite ver movimientos de una cuenta.
- CRUD de usuarios.
 - Permite añadir, eliminar y editar usuarios.
 - Se comprueba el formato del dni.
- Cierre de cuentas.
 - Se comprueba el formato de cuenta.
 - Se comprueba que la cuenta exista.
 - Se comprueba que la cuenta no tenga fondos.

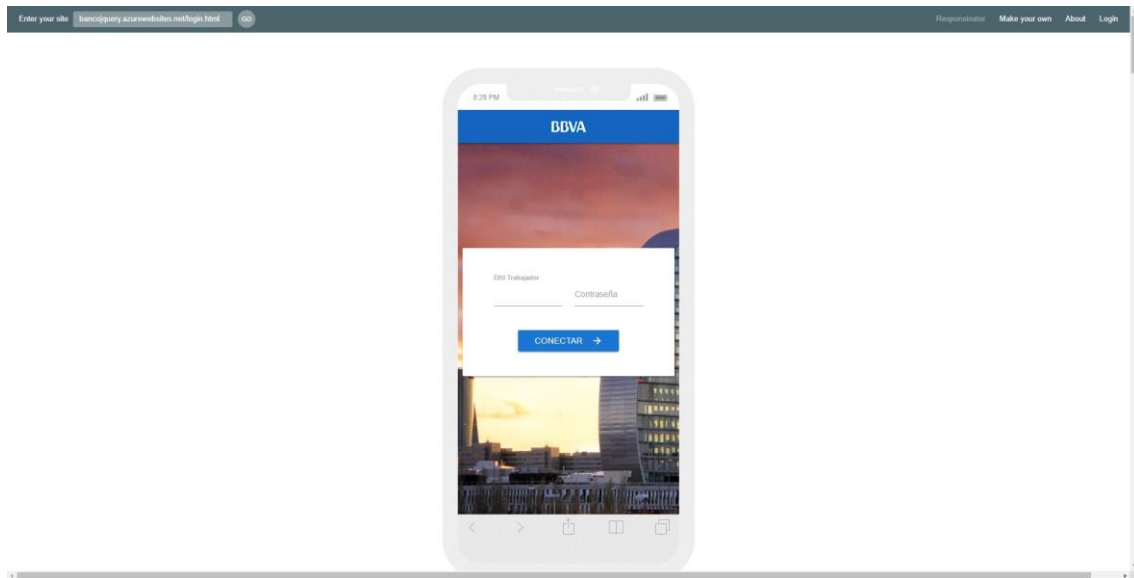


3. PRUEBAS

La aplicación fue desplegada a azure, en la url <https://bancojquery.azurewebsites.net>. De dicho modo, se han realizado pruebas en distintos apartados utilizando la web de modo público.

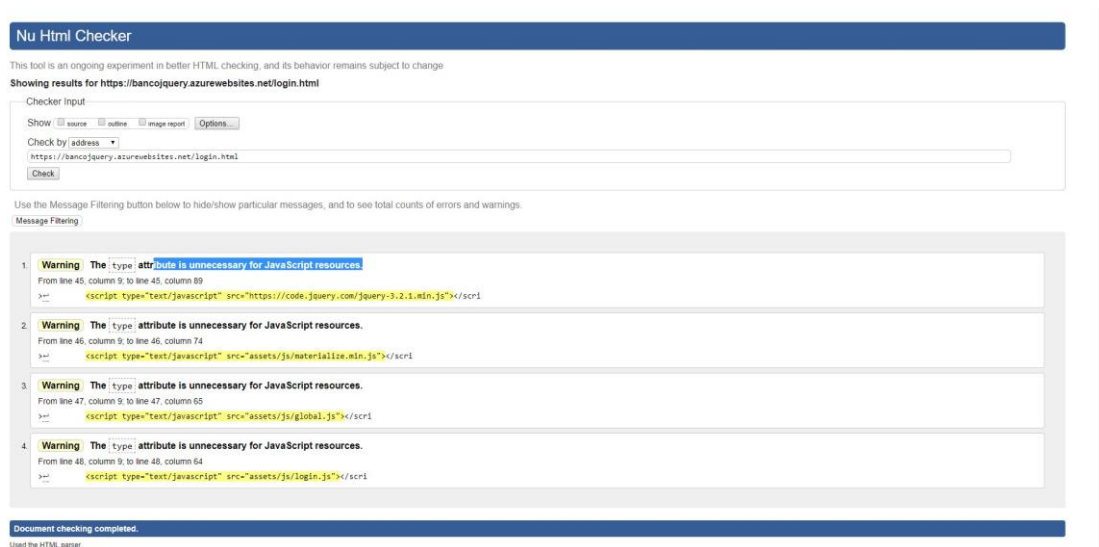
3.1. PRUEBAS DE RESPONSIVE

Las pruebas de responsive fueron satisfactorias en todos los dispositivos... [Prueba](#).



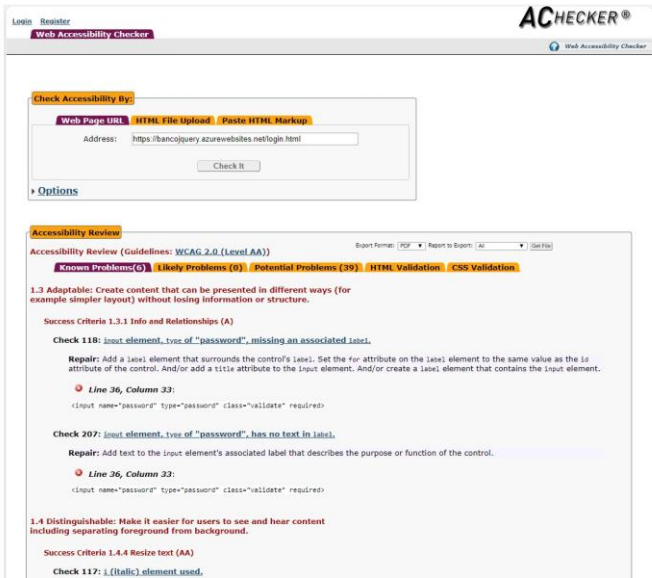
3.2. PRUEBAS DE HTML5

No presentó ningún error. Asimismo, las alertas fueron subsanadas posteriormente.



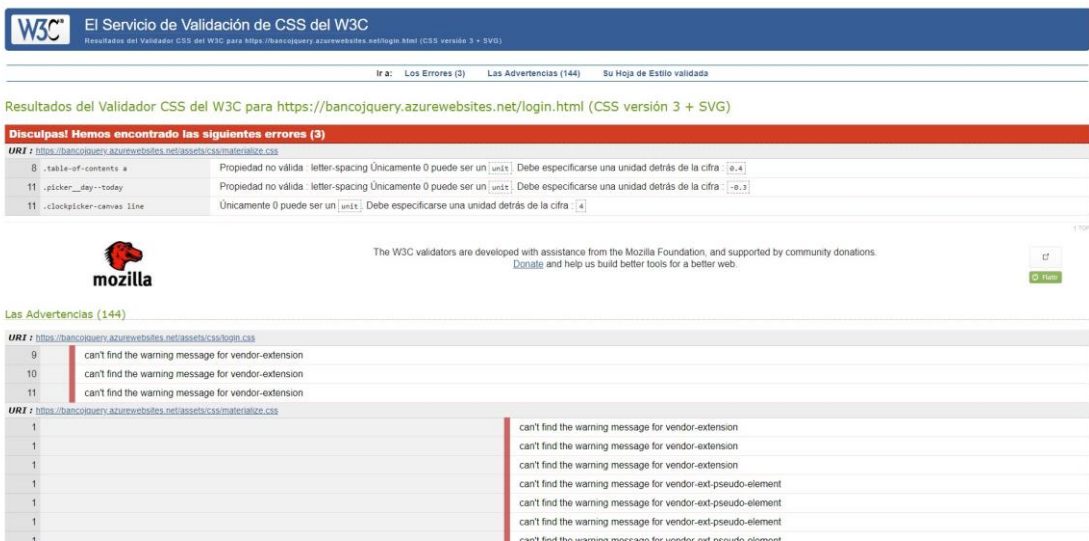
3.3. PRUEBAS DE ACCESIBILIDAD

Se presentaron algunos errores derivados del framework utilizado, pero el resto fueron subsanados.



3.4. PRUEBAS DE CSS

Tal y como se puede apreciar, se lanzaron muchas advertencias y algunos errores. Los errores fueron corregidos, pero no se han resuelto las advertencias otorgadas por el framework, ya que supondría una nueva rama de desarrollo y la incapacidad de actualizar el mismo.



4. BASE DE DATOS

La base de datos se ha modificado para facilitar las labores de administración y claridad de la aplicación. Del mismo modo, se han añadido varios campos para otorgar aún más funcionalidades a la aplicación. El modelo final es el siguiente:

- **api_calls**(id,api_key,call_timestamp,call_page)
- **api_keys**(key,description)
- **clientes**(dni,nombre,direccion,telefono,email,fecha_nacimiento,fecha_conexion,contenid_o_cuentas,saldo,pin)
- **clientes_cuentas**(dni,numero_cuenta)
- **cuentas**(numero_cuenta,saldo)
- **movimientos**(numero_cuenta,fecha,descripcion,importe)
- **usuarios**(id,nombre,apellidos,dni,pass,token)

5. SEO

El diseño ha implementado medidas de especial atención al SEO. Por ende, se han agregado etiquetas a las cabeceras para posicionar la página lo más alto en las búsquedas para “bancos” o “bbva”. Se han priorizado los buscadores de Google y Bing, al ser los más utilizados.



Pueden verse todas las etiquetas utilizadas y sus usos desde la página <https://www.40defiebre.com/guia-seo/como-hacer-web-optimizada-seo/>.

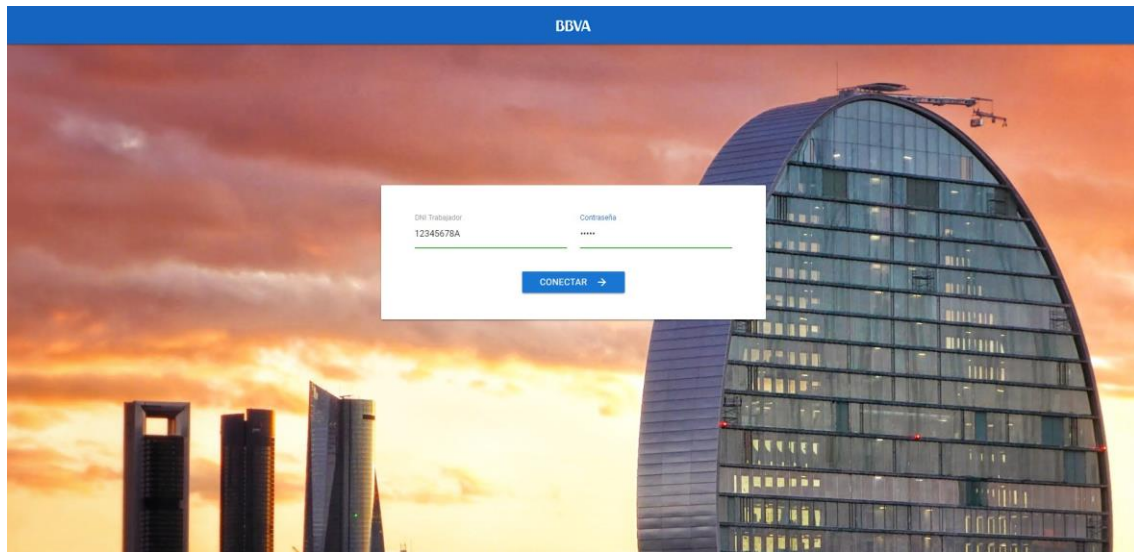
6. ASPECTO Y PÁGINAS DE LA APLICACIÓN

6.1. SITEMAP

- **index.html**: Página de redirección.
- **login.html**: Página de conexión
- **home.html**: Página de inicio tras conexión.
- **close_account.html**: Página para el cierre de cuentas.
- **list_transactions.html**: Lista (y filtra) movimientos.
- **open_account.html**: Apertura de cuentas.
- **transactions.html**: Ingresos y reintegros.
- **users.html**: CRUD de empleados del banco.

Aplicación bancaria

6.2. GALERÍA DE IMÁGENES



CONEXIÓN DE PERSONAL.



PANTALLA DE INICIO TRAS LA CONEXIÓN.

Apertura de cuentas Cierre de cuentas Ingresos y reintegros BBVA Listado de movimientos CRUD de Personal Andrei Garcia

Apertura de cuentas

ASOCIAR CLIENTE →

Número de cuenta

ABRIR CUENTA →

PANTALLA BÁSICA DE APERTURA DE CUENTAS.

Apertura de cuentas Cierre de cuentas Ingresos y reintegros BBVA Listado de movimientos CRUD de Personal Andrei Garcia

Información de la cuenta

La cuenta no se puede cerrar. Para poder cerrar la cuenta, esta no debe tener fondos.

Número de cuenta: 0000000033 Saldo: \$ 899588 Número de titulares: 1

Santiago Alonso

DNI: 1111111128 Teléfono: 913170047

CANCELAR CERRAR CUENTA

COMPROBAR CUENTA →

PANTALLA EXTENDIDA DE CIERRE DE CUENTAS, AL PROCEDER A SU CIERRE.

Apertura de cuentas Cierre de cuentas Ingresos y reintegros BBVA Listado de movimientos CRUD de Personal Andrei Garcia

Ingresos y reintegros

Ingreso Reintegro

Número de cuenta Cantidad de la operación

Descripción de la operación

CONFIRMAR →

PANEL DE INGRESOS Y REINTEGROS.

Aplicación bancaria

Apertura de cuentas

Cierre de cuentas

Ingresos y reintegros

BBVA

Listado de movimientos

CRUD de Personal

Andrei Garcia

Listado de movimientos

Número de cuenta

Fecha de inicio

Fecha final

FILTRAR

Mostrar

10

registros

Buscar

Número de cuenta	Fecha	Descripción	Importe
0000000011	9/3/2018 15:46:13	;))	-300€
0000000011	9/3/2018 15:49:15	;))	-300€
0000000022	9/3/2018 13:09:47	Reintegro de Endesa.	-500€
0000000022	9/3/2018 15:47:08	;))	-300€
0000000022	9/3/2018 15:49:08	;))	-300€
0000000033	9/3/2018 13:00:49	Ingreso de Endesa.	500€

Número de cuenta

Fecha

Descripción

Importe

Mostrando registros del 1 al 6 de un total de 6 registros

Anterior

1

Siguiente

BBVA Manager

Gestión rápida y eficaz del banco para el personal.

Andrei Garcia

Ingresos y reintegros

Listado de movimientos

Apertura de cuentas

Cierre de cuentas

LISTADO DE MOVIMIENTOS CON FILTROS.

Apertura de cuentas

Cierre de cuentas

Ingresos y reintegros

BBVA

Listado de movimientos

CRUD de Personal

Andrei Garcia

CRUD de personal

AÑADIR USUARIO

Mostrar

10

registros

Buscar

ID	DNI	Nombre	Apellidos	
1	12345678A	Andrei	Garcia	ACCIONES
4	12345687B	USUARIO	CHULO	ACCIONES
5	12345678C	23213	12	ACCIONES
10	12345678F	TEST	TEST	ACCIONES

ID

DNI

Nombre

Apellidos

Mostrando registros del 1 al 4 de un total de 4 registros

Anterior

1

Siguiente

BBVA Manager

Gestión rápida y eficaz del banco para el personal.

Andrei Garcia

Ingresos y reintegros

Listado de movimientos

Apertura de cuentas

Cierre de cuentas

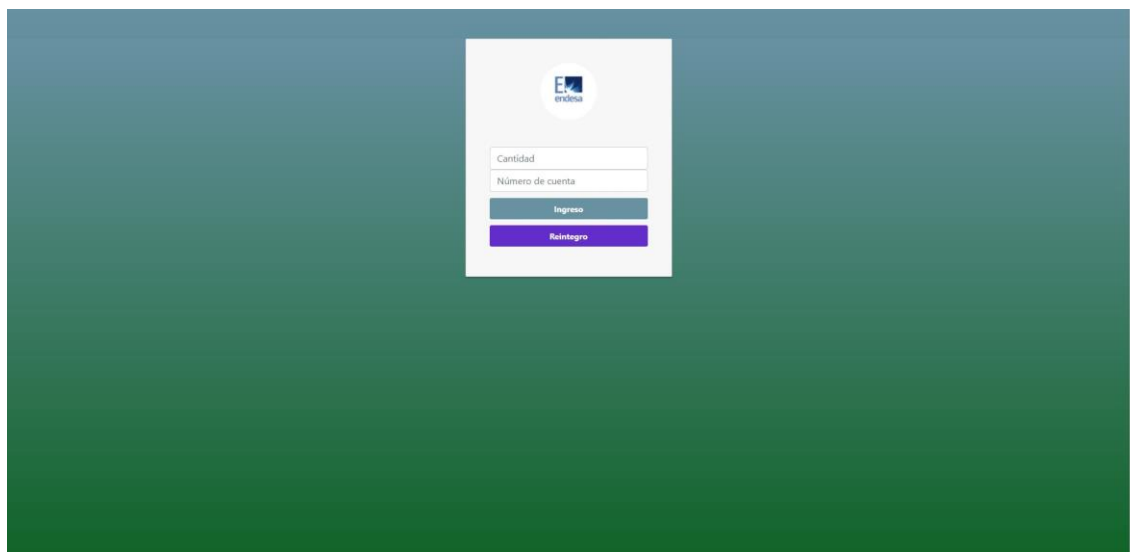
AÑADIR, MODIFICAR O ELIMINAR PERSONAL DESDE EL LISTADO.

7. ANEXOS

7.1. APLICACIÓN EXTERNA (NO OFICIAL, CON ACCESO API)

Usa la autenticación por api key anteriormente mencionada. Consiste en un cliente de terceros (realizado con Bootstrap, no responsive, sin estándares, con PHP) en el cual se enseña cómo no se deben hacer las cosas, ya que esta aplicación es realmente difícil de mantener al mezclar código JavaScript, CSS, HTML y PHP en el mismo documento.

La función de esta aplicación es tan sólo reintegrar/ingresar. Es útil para cubrir recibos de terceros (como por ejemplo, Endesa cobrando a un cliente).



7.2. CONEXIÓN DE CLIENTES

Se permite la conexión de clientes, aunque actualmente no está disponible, todo el sistema está preparado para ello. Es por esto que se agregó el campo "pin" a la tabla usuarios, y se han de desplegar unas nuevas rutas al REST.

8. COPYRIGHT Y COPYLEFT

Agradecimientos a Microsoft, Adobe y Google por el software proporcionado (Windows, Photoshop y Buscador, respectivamente).

El proyecto y sus imágenes están protegidas por derechos de autoría. Se prohíbe su distribución, copia y compartición ya que son ficheros privados de la empresa BBVA.

Aplicación bancaria

Trabajo colaborativo de DES y DIW MARZO 2018

