

Anexo IV: Documentación Técnica

APLICACIÓN PARA LA GESTIÓN ECONÓMICA PERSONAL



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Autor:

Victor Fagúndez Poyo

Tutor:

Dionisio Tomás Rodríguez Barrios

Fecha de Adjudicación: Febrero 2020

Fecha de Presentación: Julio 2020

Grado en Ingeniería Informática en Sistemas de Información

Escuela Politécnica Superior de Zamora



Historial de Revisiones

<i>Versión</i>	Fecha	Descripción
<i>1.0.0</i>	05.04.20	Versión inicial del documento
<i>1.1.0</i>	15.06.20	Añadido diario de actividades

CONTENIDO	
1.	Introducción..... 6
2.	Backend 7
2.1.	Distribución del código Fuente..... 7
2.2.	Documentación de la api 9
2.2.1.	Ruta de acceso 11
2.2.2.	Creación y Acceso de Usuarios 12
2.2.2.1.	Registro..... 12
2.2.2.2.	Inicio de sesión 13
3.	Frontend..... 15
3.1.	Documentación de componentes 17
3.1.1.	Componentes de Navegación 17
3.1.2.	Componentes de Información..... 18
3.1.3.	Componentes de acceso a información 18
3.1.4.	Componentes de creación y modificación..... 19
4.	Diario de actividades 20
4.1.	Primeros meses 20
4.2.	Marzo..... 21
4.3.	Abril..... 23
4.4.	Mayo..... 25
4.5.	Junio..... 27
4.6.	Total..... 28
5.	Abreviaturas 29
6.	Bibliografía..... 30

TABLA DE ILUSTRACIONES

Ilustración 1. Estructura del código de la API.....	8
Ilustración 2. Página inicio documentación de la API.....	9
Ilustración 3. Métodos para acceder a Cuenta.....	10
Ilustración 4. Petición Cuentas con error.....	10
Ilustración 5. Prueba de obtención de Cuentas del usuario de Prueba	11
Ilustración 6. Preparación de los Headers de la petición signup	12
Ilustración 7. Introducimos los parámetros de la petición signup	13
Ilustración 8. Resultado de la petición de creación de un nuevo usuario.....	13
Ilustración 9. Parámetros de Inicio de sesión	14
Ilustración 10. Resultado del inicio de sesión	14

TABLA DE TABLAS

Tabla 1.Estructura del codigo del frontend	16
Tabla 2. Componentes de Navegación	17
Tabla 3. Componentes de Información.....	18
Tabla 4. Componentes de acceso a información	18
Tabla 5. Componentes de creación y modificación.....	19
Tabla 6. Actividades primeros meses	20
Tabla 7. Actividades Marzo.....	22
Tabla 8. Actividades marzo	24
Tabla 9. Actividades mayo	26
Tabla 10. Actividades Junio	27
Tabla 11. Tiempo total.....	28

1. INTRODUCCIÓN

Como ya comentamos en el anexo del diseño del sistema, el sistema se divide en dos partes principales el backend y el frontend.

Para la documentación del backend en Yii2 lo dividiremos a su vez en otras dos partes, una parte dedicada a la estructura interna del sistema de servicios web RESTFUL y otra para documentar las peticiones REST disponibles.

En la parte dedicada a la estructura interna del sistema RESTFUL usaremos el programa Doxygen para que genere la documentación a partir de los ficheros PHP. Para el caso de documentar las peticiones REST disponibles usaremos la extensión de Yii2 “yii2-rest-api-doc” el cual es un generador de documentación para aplicaciones Yii2 REST basado en los puntos API finales definidos y en los comentarios de sus acciones.

2. BACKEND

Dado el uso de la herramienta Doxygen la documentación del código fuente no se encuentra en este anexo, pero se adjunta en formato digital junto con el resto de los anexos. Para acceder al documento solo hay que abrir el fichero index.html.

2.1. DISTRIBUCIÓN DEL CÓDIGO FUENTE

Para la creación de un sistema modutable y ampliable para futuras versiones se ha decidido seguir las recomendaciones de la guía de Yii2 para la creación de una API.

“Una buena API ha de ser versionada: los cambios y las nuevas características son implementadas en las nuevas versiones del API, en vez de estar continuamente modificando sólo una versión. Al contrario que en las aplicaciones Web, en las cuales tienes total control del código de ambas partes lado del cliente y lado del servidor, las APIs están destinadas a ser usadas por los clientes fuera de tu control. Por esta razón, la compatibilidad hacia atrás (BC Backward compatibility) de las APIs ha de ser mantenida siempre que sea posible. Si es necesario un cambio que puede romper la BC, debes de introducirla en la nueva versión del API, e incrementar el número de versión. Los clientes que la usan pueden continuar usando la antigua versión de trabajo del API; los nuevos y actualizados clientes pueden obtener la nueva funcionalidad de la nueva versión del API.”

[1]

En base a esto la estructura del código fuente del sistema RESTFUL es similar al siguiente esquema, en el cual se han eliminado carpetas propias del framework:

```
suma/  
  config/  
  controllers/  
    SiteController.php  
  models/  
    LoginForm.php  
    User.php  
  modules/  
    v1/  
      controllers/  
        CategoriaController.php  
        CategoriaMovimientoController.php  
        CuentaController.php  
        EmpresaController.php  
        EmpresaProductoController.php  
        MovimientoController.php  
        ProductoController.php  
        UserController.php  
        UsuarioController.php  
        UsuarioCuentaController.php  
      models/  
        Categoria.php  
        CategoriaMovimiento.php  
        CategoriaMovimientoQuery.php  
        CategoriaQuery.php  
        Cuenta.php  
        CuentaQuery.php  
        Empresa.php  
        EmpresaProductoQuery.php  
        EmpresaQuery.php  
        LoginForm.php  
        Movimiento.php  
        MovimientoQuery.php  
        Producto.php  
        ProductoQuery.php  
        SingupForm.php  
        Usuario.php  
        UsuarioCuenta.php  
        UsuarioCuentaQuery.php  
      resourcers/  
        CategoriaResource.php  
        CuentaResource.php  
        MovimientoResource.php  
        ProductoResource.php  
        UserResource.php
```

Ilustración 1. Estructura del código de la API

Observando esta estructura de carpetas y archivos, destacan en las carpetas mas externas de la estructura los modelos y controladores, propios del Modelo Vista Controlador (MVC) usado por el framework Yii2, encargadas de los aspectos más generales del sistema.

Posteriormente se agrupan dentro de la carpeta modules las diferentes versiones de la API de tal modo que cada vez que se añadan nuevas funcionalidades se añada una nueva

versión con su carpeta “vX”, siendo X el numero de la versión. Esta estructura de carpetas permite incorporar nuevas funcionalidades a la par que las antiguas versiones sigan funcionando, tal como establece las recomendaciones de la guía de Yii2 para APIs.

2.2. DOCUMENTACIÓN DE LA API

Como ya comentamos en la introducción de este anexo, para la documentación de la API se ha hecho uso de la extensión del framework Yii2 “yii2-rest-api-doc” el cual es un generador de documentación para aplicaciones Yii2 REST.

Para acceder a esta documentación, suponiendo que se ha seguido la guía de instalación del anexo 4 manual de usuario, estará en la ruta <http://localhost/suma/web/documentation>.

Al acceder a esta pagina se nos mostrara la siguiente pantalla.



The screenshot shows the initial page of the API documentation. At the top, there is a dark header bar with the text "My Application" on the left and navigation links "Home", "About", "Contact", and "Login" on the right. Below the header, there are two input fields: "Base URL" and "Authentication Token". Underneath these fields is a list of API endpoints, each in its own box: "Cuenta", "Movimiento", "Categoria", and "Producto".

Ilustración 2. Página inicio documentación de la API

Si seleccionamos una de las diferentes opciones podemos ver los diferentes métodos para acceder a él y una descripción de estos.

Cuenta	
POST	/v1/cuenta - Crear una nueva cuenta
GET	/v1/cuenta - Obtener cuentas del usuario autenticado
PUT	/v1/cuenta/{id} - Actualizar los datos de una cuenta
DELETE	/v1/cuenta/{id} - Eliminar cuenta seleccionada
GET	/v1/cuenta/{id} - Ver datos de la cuenta seleccionada

Ilustración 3. Métodos para acceder a Cuenta

Si pulsamos en uno de estos métodos podemos probarlo para lo cual necesitaremos introducir un token de acceso, de lo contrario nos dará un error, y una dirección base URL de nuestro proyecto.

Cuenta

POST /v1/cuenta - Crear una nueva cuenta

GET /v1/cuenta - Obtener cuentas del usuario autenticado

Obtener cuentas del usuario autenticado

Send

Loading...

Action: GET /v1/cuenta
Status: 404 (Not Found)
Headers:
Body:

Ilustración 4. Petición Cuentas con error

Para poder probar, a continuación, se proporcionan unas claves de pruebas, siempre y cuando se haya instalado la base de datos con los datos de prueba.

- Authentication Token: `ugL6M4TGeK2Gleiqv093az1ecsmO5cyX`
- Base URL: <http://localhost/suma/web>

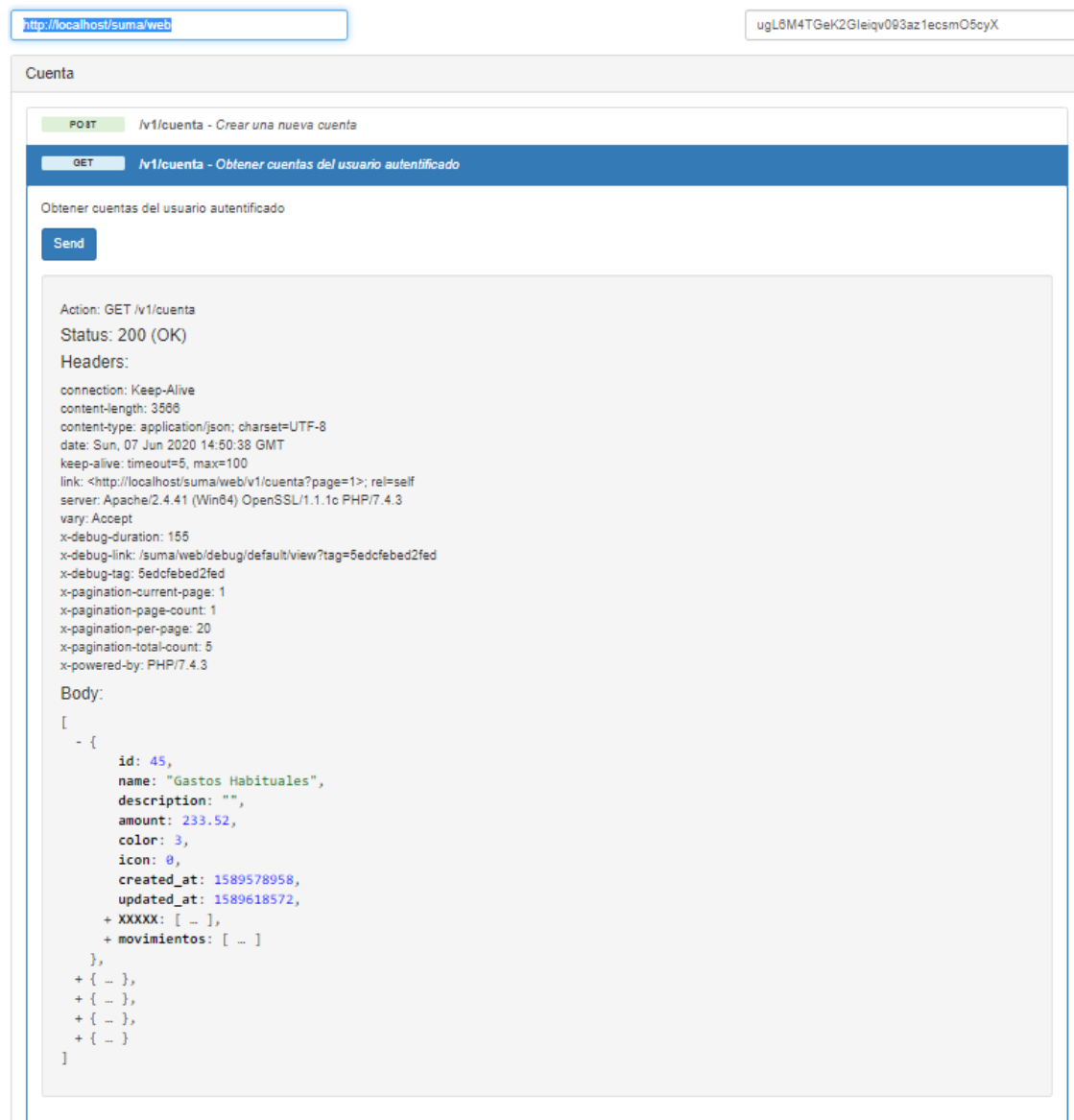


Ilustración 5. Prueba de obtención de Cuentas del usuario de Prueba

2.2.1. RUTA DE ACCESO

Como podemos observar las rutas de acceso se forman a partir de la ruta base donde tengamos instalado el sistema, mas la versión de la API que queramos usar, mas el elemento al que queremos acceder. Podría seguirse la siguiente formula:

Ruta de Instalación / vX / elemento

Siendo X el numero de la versión.

2.2.2. CREACIÓN Y ACCESO DE USUARIOS

Siguiendo la formula anterior, para poder acceder a los elementos de creación y registro de usuarios hay que seguir las siguientes rutas:

2.2.2.1. REGISTRO

Verbo	Ruta de acceso	Parámetros	
POST	v1/user/signup	Username	El nombre del usuario
		Password	La contraseña del usuario
		password_repeat	La repetición de la contraseña

A continuación, podemos ver un ejemplo usando el programa POSTMAN, el cual nos permite hacer peticiones de forma sencilla:

El primer paso es preparar los headers para la prueba, ponemos el key Content-Type con el value application/json

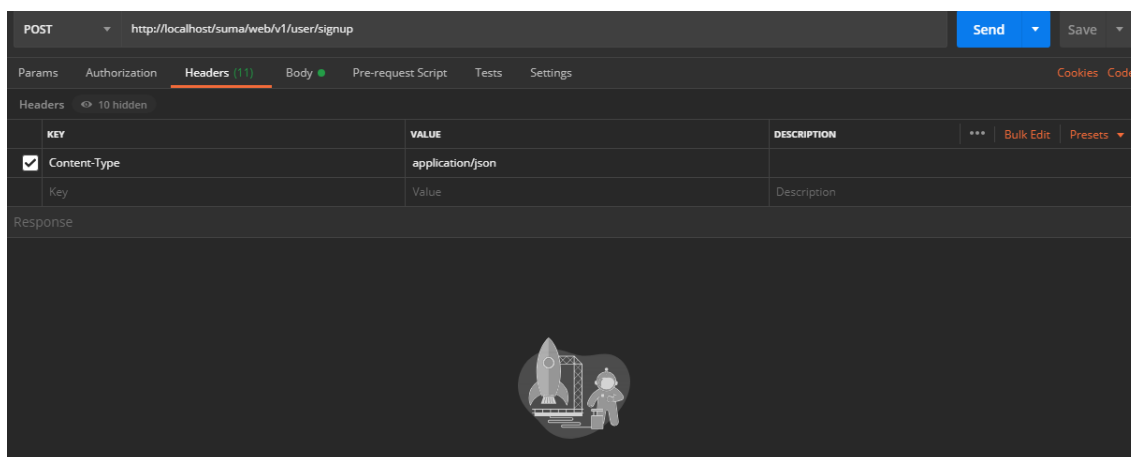


Ilustración 6. Preparación de los Headers de la petición signup

Posteriormente tenemos que pasarle los parámetros correspondientes a esta petición

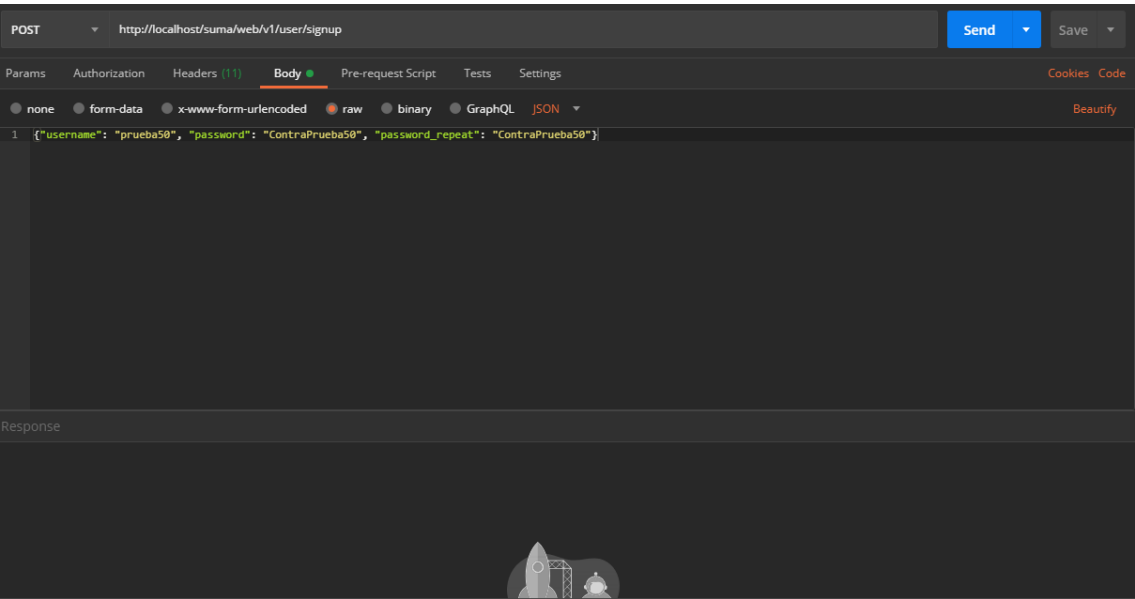


Ilustración 7. Introducimos los parámetros de la petición signup

Esto nos proporcionará como resultado los datos del usuario

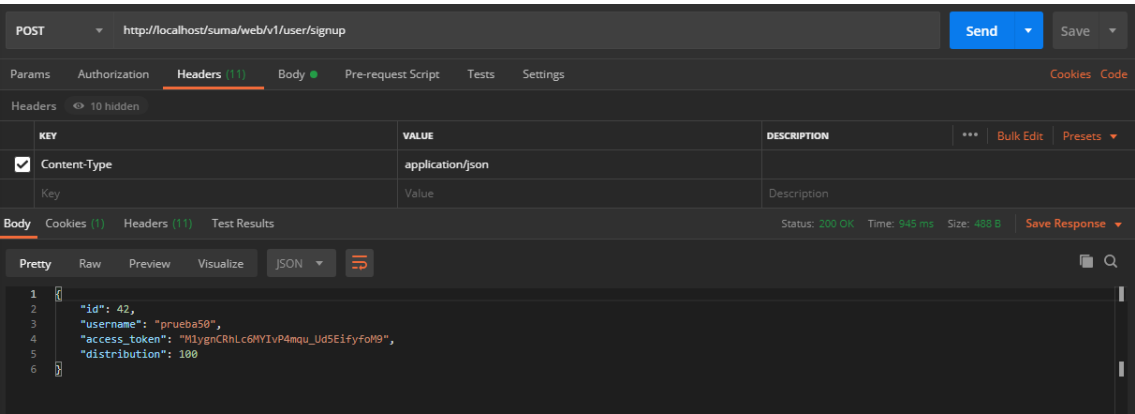


Ilustración 8. Resultado de la petición de creación de un nuevo usuario

2.2.2.2. INICIO DE SESIÓN

Verbo	Ruta de acceso	Parámetros	
POST	v1/user/login	Username	El nombre del usuario
		Password	La contraseña del usuario

A continuación, podemos ver un ejemplo, con los datos del usuario creado en el ejemplo anterior.

En primer lugar, tenemos que adecuar los parámetros

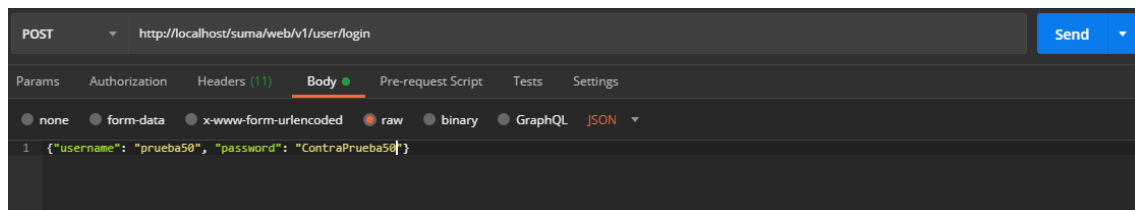


Ilustración 9. Parámetros de Inicio de sesión

Al hacer la petición nos devuelve como resultado los datos del usuario junto con su Access_token lo que nos permitirá hacer el resto de las peticiones.

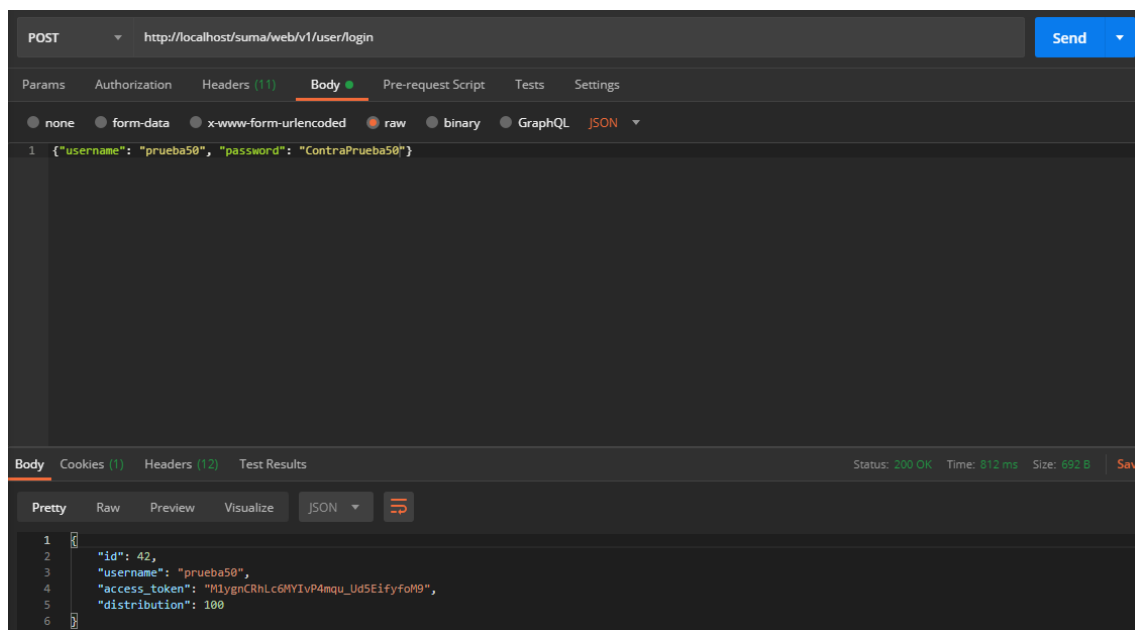


Ilustración 10. Resultado del inicio de sesión

3. FRONTEND

En cuanto al sistema frontend, desarrollado en el framework javascript React JS, este se basa en componentes reutilizables.

Dentro del código propio del framework React, la estructura del código de los componentes es la siguiente dentro de la carpeta src/.



Tabla 1.Estructura del código del frontend

En esta estructura podemos observar la carpeta vistas, alrededor de la cual se estructura el resto del código, la cual recoge las diferentes vistas de la aplicación. Estas vistas hacen uso de los componentes, los cuales se alojan en la carpeta components, para formar sus interfaces.

La carpeta classes guarda dos archivos que facilitan la reutilización de código repetitivo de los componentes, como elementos de elección de interfaz y datos de los usuarios. En la carpeta assets se recogen los elementos visuales de ayuda al usuario y las fuentes usadas.

3.1. DOCUMENTACIÓN DE COMPONENTES

3.1.1. COMPONENTES DE NAVEGACIÓN

Los componentes de navegación sirven para acceder a las diferentes páginas, formadas, a su vez, por otros componentes.

Componente	Funcionalidad
BottomNavigation.js	Proporciona una barra de navegación inferior en versiones móviles que da acceso a las diferentes vistas de la página.
Navbar.js	Barra superior de navegación que da acceso a las diferentes vistas de la página en la versión para ordenador.
Informacion.js	Componente que recoge y forma la estructura de la vista principal del programa con el resto de los componentes.
InformacionCuentas.js	Componente que recoge y forma la estructura de la vista de Cuentas del programa con el resto de los componentes.
InformacionMovimiento.js	Componente que recoge y forma la estructura de la vista de Movimientos del programa con el resto de los componentes.
InformacionUsuario.js	Componente que recoge y forma la estructura de la vista de Usuario del programa con el resto de los componentes.

Tabla 2. Componentes de Navegación

3.1.2. COMPONENTES DE INFORMACIÓN

Los componentes de información muestran de manera visual la información relativa al usuario, ingresos, gastos, reparto de cuentas, etc.

Componente	Funcionalidad
GraficoDistribucionCuentas.js	Componente que muestra en un gráfico circular la distribución real del dinero del usuario entre sus diferentes cuentas.
GraficoDistribucionIngresos.js	Componente que muestra en un gráfico circular la distribución del % de ingresos entre las diferentes cuentas del usuario.
GraficoIngresosGastos.js	Componente que muestra en un grafico de barras los gastos y la cantidad ahorrada correspondiente en base a sus ingresos a lo largo de los meses de un año.
UltimoMes.js	Componente que muestras los ingresos y gastos del mes actual.

Tabla 3. Componentes de Información

3.1.3. COMPONENTES DE ACCESO A INFORMACIÓN

Componente	Funcionalidad
InfoMovimiento.js	Componente que muestra la información concreta de un movimiento.
AvatarUsuario.js	Componente que muestra el avatar del usuario.
CuentasEnTarjetas.js	Componente que muestra en forma de tarjeta las cuentas de las que dispone el usuario.
TarjetaCuenta.js	Componente que muestra información de una cuenta del usuario en forma de tarjeta.
DatosUsuario.js	Componente que muestra el numero total de cuentas y movimientos de los que dispone un usuario.

Tabla 4. Componentes de acceso a información

3.1.4. COMPONENTES DE CREACIÓN Y MODIFICACIÓN

Componente	Funcionalidad
FloatButton.js	Componente de botón flotante desde el cual se pueden crear nuevos movimientos y nuevas cuentas.
ModificarCuenta.js	Componente que permite cambiar los datos de una cuenta concreta.

Tabla 5. Componentes de creación y modificación

4. DIARIO DE ACTIVIDADES

4.1. PRIMEROS MESES

Fecha	Actividad	Duración (min)	Observaciones
21.11.19	Boceto inicial de la interfaz de la aplicación	90	
25.12.19	Búsqueda de guía de estilo para la interfaz	120	En páginas como behance.net
17.02.20	Creación del proyecto react	80	Usando Create React app
23.02.20	Se muestran las cuentas a partir de un fichero json en la aplicación	75	
24.02.20	Se ha creado la barra de navegación inferior	180	
25.02.20	Se ha solucionado errores y warnings y se ha empezado a crear un menú desplegable para el usuario	35	
25.02.20	Solucionado error en la barra de navegación	25	
Total Mes		605	10,0833 h

Tabla 6. Actividades primeros meses

4.2. MARZO

02.03.20	Actualizar el titulo de la barra de navegación en función de la página actual	25	
02.03.20	Añadir botón de opciones al componente cuenta	30	
03.03.20	Modelar y dar estilo al componente cuenta	90	
04.03.20	Ajustes en el tamaño del componente cuenta	25	
05.03.20	Modelar y dar estilo al componente movimiento	120	
06.03.20	Solucionar problema CORS en las peticiones fetch al servidor	60	Se ha incorporado la variable proxy en las variables del proyecto
07.03.20	Añadir nuevos campos a la base de datos y adaptarla a las modificaciones del componente cuenta	30	
08.03.20	Añadidos iconos y colores al componente cuenta	60	
10.03.20	Añadidos iconos y colores al componente movimiento	45	
10.03.20	Añadir nuevos campos a la base de datos y adaptarla a las modificaciones del componente movimiento	70	
11.03.20	Modelar el Login	180	
11.03.20	Creación del botón para pasar entre el login y el registro	45	
13.03.20	Integración de la librería React Content Loader	55	Para mejorar la visualización de la carga de elementos
13.03.20	Corregir superposición de elementos con la barra superior y la navegación	30	
13.03.20	Integración de la librería react fade in para incluir una animación al cargar los elementos	55	
19.03.20	Estructurar backend en módulos y versiones según las recomendaciones de Yii2	120	
20.03.20	Login y registro en backend	300	
21.03.20	Login y registro en frontend	240	

24.03.20	Hacer que las peticiones funcionen con token	60	
26.03.20	Hacer que se devuelvan solo los datos del usuario	140	
Total mes		1780	29,66667 h

Tabla 7. Actividades Marzo

4.3. ABRIL

05.04.20	Documento análisis de requisitos versión 1.0.0	360	
09.04.20	Mostrar todas las cuentas del usuario con estilo Material-ui	220	
12.04.20	Mostrar Listado de movimientos de cada cuenta	150	
13.04.20	Imágenes de información cuando el usuario no ha seleccionado ningún elemento	84	
15.04.20	Ahora se actualiza el valor de la cuenta con cada nuevo movimiento	125	
16.04.20	Se guarda el porcentaje aportado por el usuario a la cuenta en usuariocuenta	240	
17.04.20	Información de cada uno de los movimientos	133	
19.04.20	Eliminar un movimiento específico	150	
19.04.20	Ahora la lista de cuentas son botones	65	
20.04.20	Botones para regresar entre movimientos y cuentas	85	
21.04.20	Adaptación a pantallas móviles y correcto funcionamiento	125	
22.04.20	Crear nueva cuenta y elegir % de ingresos	254	
23.04.20	Se muestra la información del movimiento	90	
23.04.20	Se ha añadido un botón de retroceso y un botón de eliminar movimiento	100	
24.04.20	Crear un nuevo gasto y elegir cuenta asociada	140	
24.04.20	Incluida selección del color al crear cuenta	80	
25.04.20	Animación de carga al cargar las cuentas del usuario	35	
26.04.20	Añadida opción de ver todos los movimientos	85	
26.04.20	Añadido añadir nuevo gasto	200	
27.04.20	Añadido Grafico para visualizar % de ingresos asociado a cada cuenta	114	

29.04.20	Se ha incluido la extensión yii2-rest-api-doc	45	
Total Mes		2880	48h

Tabla 8. Actividades marzo

4.4. MAYO

02.05.20	Solución del error con el token de acceso del usuario	25	
02.05.20	Añadido registro de nuevos usuarios	125	
03.05.20	Solucionado error al crear nuevo usuario	70	
07.05.20	Añadir nuevos controladores en el backend	120	
09.05.20	Corregido Grafico para visualizar % de ingresos asociado a cada cuenta para móviles	50	
11.05.20	Añadido modificar cuenta	114	
13.05.20	Añadida actualización del "amount" de la cuenta cuando se elimina un movimiento	78	
13.05.20	Añadido grafico con gastos y ahorros de cada mes en el Dashboard	145	
14.05.20	Corregido error al crear un nuevo gasto	68	
15.05.20	Solucionado error con la paginación de los elementos ahora devuelve todos los movimientos	36	
15.05.20	Solucionado problema con % en el gráfico de distribución de ingresos en cuentas	45	
15.05.20	Añadidos gráficos de distribución de ingresos y de cuenta en la pantalla principal	78	
15.05.20	Últimos retoques a la página principal	35	
16.05.20	Corregido error que daba problemas al visualizar el componente último mes en el móvil	15	
20.05.20	Añadida vista de usuario	240	
20.05.20	Añadidas tarjetas de cuentas y cerrar sesión en la vista de usuario en móvil	45	
20.05.20	Añadida acción de total que devuelve el número total de	85	

	movimientos y cuentas (Backend)		
28.05.20	Añadidos errores al registrarse en la aplicación	150	
Total Mes		1524	25,4 h

Tabla 9. Actividades mayo

4.5. JUNIO

01.06.20	Hacer anexo de diseño del sistema	240	
02.06.20	Hacer anexo de diseño del sistema	192	
03.06.20	Hacer anexo de diseño del sistema	165	
04.06.20	Hacer anexo de diseño del sistema	341	
05.06.20	Anexo 3 Documentación Técnica	157	
06.06.20	Anexo 3 Documentación Técnica	94	
07.06.20	Anexo 3 Documentación Técnica	274	
08.06.20	Anexo 3 Documentación Técnica	90	
08.06.20	Hacer Anexo de diseño del sistema	62	
09.06.20	Anexo 4 Manual de Usuario	312	
10.06.20	Anexo 4 Manual de Usuario	412	
11.06.20	Anexo 0 Plan del proyecto	194	
12.06.20	Memoria	338	
13.06.20	Memoria	408	
14.06.20	Memoria	166	
15.06.20	Anexo 3 Documentación Técnica	132	
Total Mes		3577	59,61 h

Tabla 10. Actividades Junio

4.6. TOTAL

Periodo	Tiempo (min)	Tiempo (horas)
Primeros Meses	605	
Marzo	1780	
Abril	2880	
Mayo	1524	
Junio	3577	
Total	10336	172 h 16 min

Tabla 11. Tiempo total

5. ABREVIATURAS

<i>Abreviatura</i>	<i>Significado</i>
<i>REST</i>	Transferencia de estado representacional (<i>Representational State Transfer</i>)
<i>API</i>	Interfaz de programación de aplicaciones (<i>Application programming interface</i>)
<i>BC</i>	Compatibilidad con versiones anteriores (<i>Backward compatibility</i>)
<i>MVC</i>	Modelo Vista Controlador

6. BIBLIOGRAFÍA

- [1] “Servicios Web RESTful: Gestión de versiones | Guía Definitiva de Yii 2.0 | Yii PHP Framework.” [Online]. Available: <https://www.yiiframework.com/doc/guide/2.0/es/rest-versioning>. [Accessed: 06-Jun-2020].