

Memoria

# APLICACIÓN PARA LA GESTIÓN ECONÓMICA PERSONAL



**VNiVERSiDAD  
DSALAMANCA**

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><b>Versión</b></i>	<b>Fecha</b>	<b>Descripción</b>
0.0.0	01.04.20	Creación del documento
1.0.0	14.06.20	Finalización del documento

CONTENIDO	
1.	Contenido y estructura de la memoria ..... 6
2.	Introducción ..... 7
3.	Objetivos del Proyecto ..... 9
3.1.	Objetivos del Software ..... 9
3.2.	Objetivos Técnicos ..... 10
4.	Conceptos Teóricos ..... 11
4.1.	Servicio Web ..... 11
4.2.	Servicios web rest ..... 11
4.3.	JSON ..... 12
4.4.	Framework ..... 12
5.	Técnicas y Herramientas ..... 13
5.1.	Proceso unificado ..... 13
5.2.	Metodología de elicitación de requisitos ..... 14
5.3.	Material-Design ..... 15
5.4.	Framework php yii2 ..... 16
5.5.	react js ..... 17
5.5.1.	Material-UI ..... 18
5.5.2.	Recharts ..... 19
5.6.	xampp ..... 20
5.7.	visual studio code ..... 21
5.8.	github ..... 22
5.9.	sublime merge ..... 22
5.10.	Adobe XD ..... 22
6.	Aspectos relevantes del desarrollo ..... 23
6.1.	Análisis de requisitos ..... 23
6.2.	Arquitectura ..... 23
6.3.	Aprendizaje ..... 24
6.4.	Diseño de la interfaz ..... 24
7.	Conclusiones y líneas de trabajo Futuro ..... 26
7.1.	Conclusiones ..... 26
7.2.	Líneas de trabajo Futuras ..... 26
8.	Abreviaturas ..... 27
9.	Bibliografía ..... 28

## TABLA DE ILUSTRACIONES

Ilustración 1. Variación trimestral del PIB de España [4] .....	7
Ilustración 2. Evolución del número de parados[5] .....	7
Ilustración 3. Ejemplo de JSON .....	12
Ilustración 4. Material Design .....	15
Ilustración 5. Yii2 Framework .....	16
Ilustración 6. React JS .....	17
Ilustración 7. Ejemplo de componente React .....	18
Ilustración 8. Material-UI .....	18
Ilustración 9. Recharts .....	19
Ilustración 10. XAMPP .....	20
Ilustración 11. Visual Studio Code .....	21
Ilustración 12. GitHub .....	22
Ilustración 13. Sublime Merge .....	22
Ilustración 14. Adobe XD .....	22
Ilustración 15. Pantalla Principal vista de ordenador .....	25
Ilustración 16. Pantalla Principal vista desde un móvil .....	25



## 1. CONTENIDO Y ESTRUCTURA DE LA MEMORIA

La memoria de este trabajo de fin de grado se estructura en base a la guía de Realización y Documentación de Proyectos de Final de Carrera[1].

En base a esto, la memoria estará dividida en dos partes bien diferenciadas, una dedicada a la descripción del proyecto, el presente documento, y otra parte dedicada a la documentación técnica del proyecto, recogida en los siguientes 5 anexos:

- **Anexo 0 Planificación del Proyecto:** Encargado de la planificación del proyecto, sus tareas y el estudio de viabilidad.
- **Anexo 1 Especificación de Requisitos:** Encargado de recoger la funcionalidad que debe cumplir el sistema
- **Anexo 2 Especificación de Diseño:** Encargado de recoger la estructura, diseño arquitectura del sistema.
- **Anexo 3 Documentación Técnica:** Manual para el programador, en el cual se recogen documentación del código.
- **Anexo 4 Manual de Usuario:** Encargado de recoger el manual de instalación del sistema y el manual de uso.

**NOTA:** Tanto esta memoria, como sus correspondientes anexos han sido entregados a través de la sede electrónica, el resto de elementos se han compartido a través de un enlace de Google Drive entregado en la entrega correspondiente a esta titulación en el curso de Studium habilitado para este fin.

## 2. INTRODUCCIÓN

Año 2008, una crisis en los mercados internacionales provoca una gran recesión en la economía global. Este hecho provocó en España, una gran crisis durante el periodo de 2009 a 2013, destruyendo entre 3 y 4 millones de empleos[2], y sufriendo una variación trimestral del PIB en el cuarto trimestre del 2008 y en el primer trimestre del 2009, del -1.58 y -2.58 respectivamente, según el Instituto Nacional de Estadística (INE)[3]. De la noche a la mañana, miles de personas perdieron sus trabajos, sus casas y sus esperanzas.

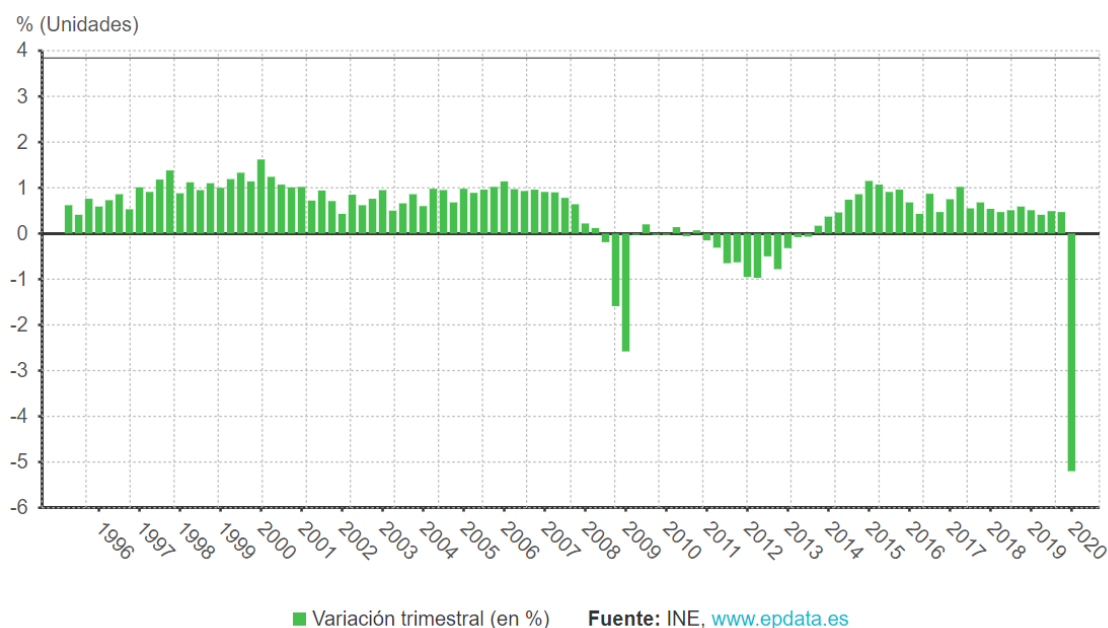


Ilustración 1. Variación trimestral del PIB de España [4]

### Evolución del número de parados

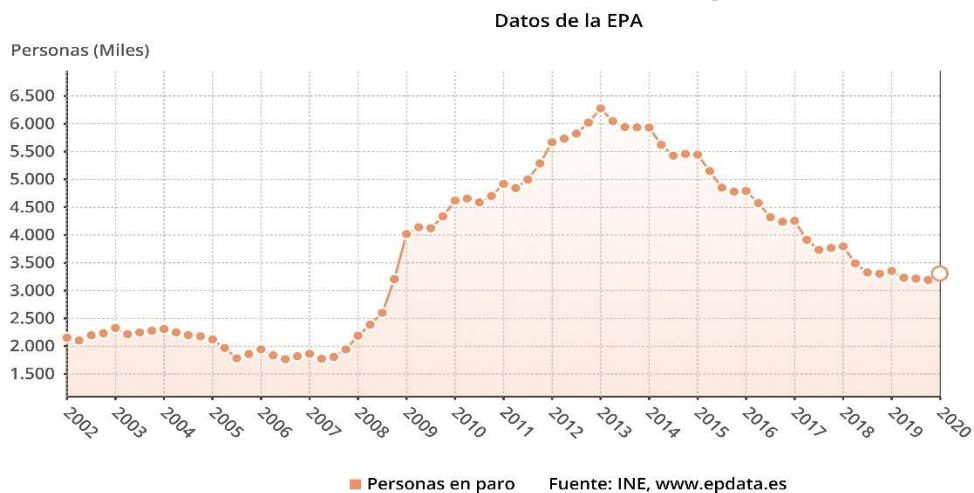


Ilustración 2. Evolución del número de parados[5]

El informe de 2018 del Ministerio de Sanidad, Consumo y Bienestar Social, Crisis económica y salud en España[6], el cual recogía que la salud mental es una de las grandes afectadas cuando se experimenta una situación financiera poco favorable. De hecho, según el estudio de Cigna, ‘360 Wellbeing Survey 2019: Well and Beyond’[7], la economía personal es, ahora mismo, la segunda fuente de estrés de los españoles. Esta sensación se conoce como estrés financiero que, de no controlarse, puede contribuir a la aparición de problemas de salud, tales como el aumento de la presión arterial, problemas de sueño, depresión u obesidad.

Según los datos obtenidos en este estudio, la calificación que otorgan los españoles a sus finanzas personales es de 55,2 sobre 100. De hecho, solo el 18% de los encuestados reconoce tener una “muy buena” o “excelente” situación financiera, porcentaje que baja 5 puntos frente a los datos obtenidos en 2018.

En base a estos datos, el objetivo del presente proyecto consiste en desarrollar una aplicación para gestionar la economía personal de los usuarios que, aunque, por si sola, no solucione los problemas financieros de los usuarios, si suponga una herramienta de gran utilidad que proporcione al usuario información relevante y fácilmente comprensible, que le ayude a tomar mejores decisiones.



### 3. OBJETIVOS DEL PROYECTO

En este apartado se muestran los objetivos que se pretenden perseguir durante este proyecto. Se puede distinguir entre los objetivos marcados por el software y los objetivos de carácter técnico.

#### 3.1. OBJETIVOS DEL SOFTWARE

El objetivo principal de este proyecto es permitir una gestión económica personal sencilla e intuitiva que ayude a los usuarios en sus finanzas personales. Para ello se han implementado herramientas visuales que permiten mostrar de forma grafica y ordenada la evolución y gestión de sus ingresos y gastos.

También se incorpora la posibilidad de crear y gestionar los ingresos mediante cuentas a las cuales se distribuirá el ingreso de forma automática en función de lo que establezca el usuario para esa cuenta.

A continuación, mostramos los objetivos del software perseguidos en este proyecto:

- Gestionar usuarios: El sistema deberá gestionar y proporcionar acceso al sistema a los diferentes usuarios.
- Gestionar Cuentas: El sistema deberá gestionar y proporcionar acceso a los datos de los usuarios correspondientes a las cuentas de los usuarios normales.
- Gestionar Movimientos: El sistema deberá gestionar y proporcionar acceso a los datos de los usuarios correspondientes a los movimientos del usuario normal.
- Visualización de datos: El sistema deberá mostrar a los usuarios información resumida en forma de gráficos.
- Interfaz sencilla y usable: El sistema proporcionará al usuario solo aquellas opciones que necesite en ese momento, permitiendo así una interfaz simple, usable y accesible.

### 3.2. OBJETIVOS TÉCNICOS

Además de los objetivos software anteriormente descritos, surgen otros objetivos técnicos y personales ligados al desarrollo como:

- Aprendizaje y uso de lenguajes, librerías y frameworks adecuados para el desarrollo de sistemas de información, útiles para alcanzar los objetivos del proyecto.
- Aplicación de conocimientos aprendidos durante la carrera.
- Adquirir conocimientos sobre nuevas tecnologías muy utilizadas actualmente.
- Afrontar el proceso completo de creación de un software.

## 4. CONCEPTOS TEÓRICOS

En este apartado se sintetizarán los conceptos teóricos necesarios para la comprensión y desarrollo para este proyecto.

### 4.1. SERVICIO WEB

Un servicio web es una tecnología para intercambiar datos entre aplicaciones mediante un conjunto de protocolos y estándares. Lo cual permite conectar distintas aplicaciones software desarrolladas con diferentes lenguajes de programación y ejecutadas en plataformas diferentes. Para poder entenderse entre las diferentes tecnologías se hace uso de estándares abiertos que faciliten la interoperabilidad.

Según W3C un servicio web “es un sistema software diseñado para soportar la interacción máquina-a-máquina, a través de una red, de forma interoperable. Cuenta con una interfaz descrita en un formato procesable por un equipo informático (específicamente en WSDL), a través de la que es posible interactuar con el mismo mediante el intercambio de mensajes SOAP, típicamente transmitidos usando serialización XML sobre HTTP conjuntamente con otros estándares web.”[8]

### 4.2. SERVICIOS WEB REST

REST describe cualquier interfaz entre sistemas que utilicen HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON que será el que utilizaremos en este proyecto, etc.)[9].

REST permite un protocolo cliente servidor sin estado, es decir, cada mensaje HTTP contiene toda la información necesaria para la correcta comprensión de la petición. Por ello, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes[9].

Además, “REST presenta un conjunto de operaciones bien definidas que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE”[9].

Estas operaciones permiten acceder a recursos que son direccionables únicamente a través de su URI.

### 4.3. JSON

JSON es formato de texto sencillo para el intercambio de datos. “Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.”[10]

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitems": [
        {
          "value": "New", "onclick": "CreateNewDoc()"
        }, {
          "value": "Open", "onclick": "OpenDoc()"
        }, {
          "value": "Close", "onclick": "CloseDoc()"
        }
      ]
    }
  }
}
```

*Ilustración 3. Ejemplo de JSON*

### 4.4. FRAMEWORK

“En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.”[11]

“Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.”[11]

## 5. TÉCNICAS Y HERRAMIENTAS

En el presente apartado se describirán las técnicas metodológicas y las herramientas de desarrollo utilizadas a lo largo del proyecto.

### 5.1. PROCESO UNIFICADO

Este proyecto se asienta sobre las bases del proceso unificado. El proceso unificado presenta las siguientes características:

- **“Conducido por casos de uso:** Los casos de uso se implementan para asegurar que toda la funcionalidad se realiza en el sistema y verificar y probar el mismo. Como los casos de uso contienen las descripciones de las funciones, afectan a todas las fases y vistas
- **Centrado en la arquitectura:** La arquitectura se describe mediante diferentes vistas del sistema. Es importante establecer una arquitectura básica pronto, realizar prototipos, evaluarla y finalmente refinarla durante el curso del proyecto
- **Iterativo e incremental:** Resulta práctico dividir los grandes proyectos en mini proyectos, cada uno de los cuales es una iteración que resulta en un incremento”[12]

El proceso Unificado se repite a lo largo de una serie de ciclos, en los que cada uno de estos ciclos consta, a su vez, de cuatro fases:

- **“Inicio:** se define el alcance del proyecto y se desarrollan los casos de negocio
- **Elaboración:** se planifica el proyecto, se especifican en detalle la mayoría de los casos de uso y se diseña la arquitectura del sistema
- **Construcción:** se construye el producto
- **Transición:** Se corrigen problemas y se incorporan mejoras sugeridas en la revisión”[12]

## 5.2. METODOLOGÍA DE ELICITACIÓN DE REQUISITOS

El objetivo de esta metodología, creada por Durán y Bernárdez, es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de elicitación de requisitos de la fase de ingeniería de requisitos del desarrollo de software.[13]

En esta metodología se distinguen dos tipos de productos: los productos entregables y los productos no entregables o internos. Los productos entregables son aquellos que se entregan oficialmente al cliente como parte del desarrollo en fechas previamente acordadas, mientras que los no entregables son productos internos al desarrollo que no se entregan al cliente.[13]

El único producto entregable definido en esta metodología es el Documento de Requisitos del Sistema (DRS). [13]

Las tareas que esta metodología recomienda para obtener los productos descritos son las siguientes:

- Tarea 1: Obtener información sobre el dominio del problema y el sistema actual.
- Tarea 2: Preparar y realizar las reuniones de elicitación/negociación.
- Tarea 3: Identificar/revisar los objetivos del sistema.
- Tarea 4: Identificar/revisar los requisitos de almacenamiento de información.
- Tarea 5: Identificar/revisar los requisitos funcionales.
- Tarea 6: Identificar/revisar los requisitos no funcionales.
- Tarea 7: Priorizar objetivos y requisitos.

### 5.3. MATERIAL-DESIGN

Material Design es una guía de diseño enfocado en la visualización del sistema operativo Android, en la web y en cualquier plataforma. “Material Design es un lenguaje visual que sintetiza los principios clásicos del buen diseño con la innovación de la tecnología y la ciencia.”[14]



*Ilustración 4. Material Design*

Material Design esta creado para proporcionar una correcta accesibilidad a los usuarios presentando formas, figuras e iconos simples y claros que facilitan el entendimiento de las metáforas utilizadas.

Para ayudar en el objetivo de crear la mejor accesibilidad posible, Material Design se ayuda de un color primario y otro secundario para representar la marca. Los temas de color están diseñados para ser armoniosos, garantizar un texto accesible y distinguir elementos y superficies de la interfaz de usuario.

## 5.4. FRAMEWORK PHP YII2

Yii es un framework genérico de programación web, lo que significa que se puede utilizar para desarrollar todo tipo de aplicaciones web en PHP. Debido a su arquitectura basada en componentes y a su sofisticada compatibilidad de caché, es especialmente apropiado para el desarrollo de aplicaciones de gran envergadura, como páginas web, foros, sistemas de gestión de contenidos (CMS), proyectos de comercio electrónico, servicios web compatibles con la arquitectura REST y muchos más.[15]



*Ilustración 5. Yii2 Framework*

Yii presente estas características:

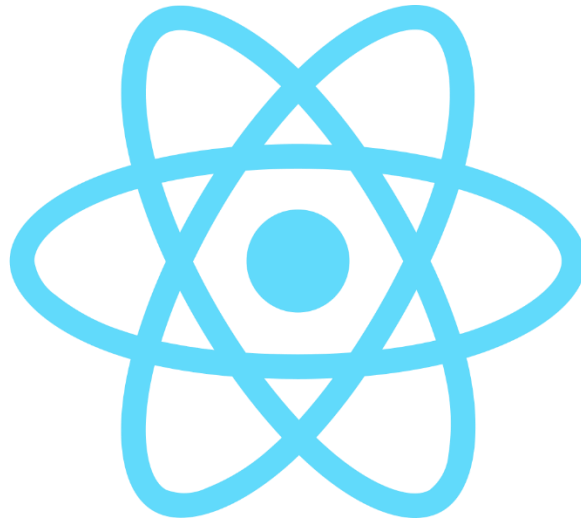
- Yii implementa el patrón de diseño MVC (Modelo-Vista-Controlador) y promueve la organización de código basada en este patrón
- Yii es un framework completo (full stack) que provee muchas características probadas y listas para usar, como los constructores de consultas y la clase ActiveRecord para las bases de datos relacionales y NoSQL junto con la compatibilidad con la arquitectura REST para desarrollar API.
- Yii es extremadamente extensible. Puedes personalizar o reemplazar prácticamente cualquier pieza de código de base, como se puede también aprovechar su sólida arquitectura de extensiones para utilizar o desarrollar extensiones distribuibles.

Estas características y sobre todo la compatibilidad con la arquitectura REST para desarrollar API ha sido una de las características esenciales para la elección de este framework para el desarrollo del proyecto.



## 5.5. REACT JS

React JS es una biblioteca JavaScript de código abierto para construir interfaces de Usuario basada en componentes. React es la Vista en un contexto en el que se use el patrón MVC (Modelo-Vista-Controlador).



*Ilustración 6. React JS*

React presenta las siguientes características:

- **Virtual DOM:** React mantiene un virtual DOM propio, en lugar de confiar solamente en el DOM del navegador. Esto deja a la biblioteca determinar qué partes del DOM han cambiado comparando contenidos entre la versión nueva y la almacenada en el virtual DOM, y utilizando el resultado para determinar cómo actualizar eficientemente el DOM del navegador.[16]
- **Las propiedades:** Las propiedades (también conocidas como 'props') pueden definirse como los atributos de configuración para dicho componente. Éstas son recibidas desde un nivel superior, normalmente al realizar la instancia del componente y por definición son inmutables.[16]
- **El Estado:** El estado de un componente se define como una representación de este en un momento concreto, es decir, una instantánea del propio componente. Existen dos tipos de componentes con y sin estado, denominados statefull y stateless.[16]
- **Ciclos de vida:** El ciclo de vida es una serie de estados por los cuales pasan los componentes statefull a lo largo de su existencia. Se pueden clasificar en tres

etapas de montaje o inicialización, actualización y destrucción. Dichas etapas tienen correspondencia en diversos métodos.

- **JSX:** React utiliza una sintaxis parecida a HTML, llamada JSX. No es necesaria para utilizar React, sin embargo, hace el código más legible, y escribirlo es una experiencia similar a HTML.

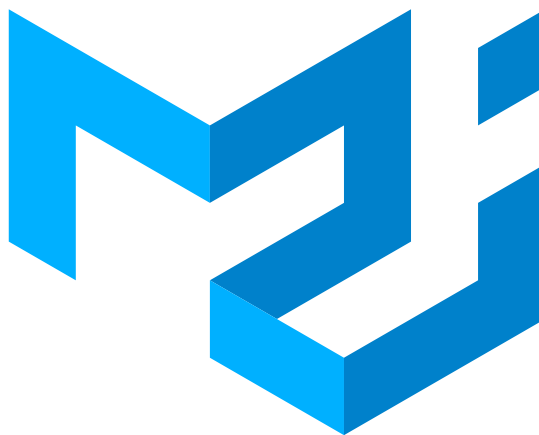
```
react.js
1 import React from 'react'; //Se importa React
2
3 class Componente extends React.Component{ //Declaración de un componente con el nombre de "Componente".
4   render(){
5     //Dentro de return se escribe en una sintaxis muy parecida a HTML
6     return(
7       <div>
8         <h1>Hola mundo</h1>
9       </div>
10    );
11  }
12 }
13
14 ReactDOM.render(<Componente />, document.getElementById('MiAppDeReact'))
15
```

*Ilustración 7. Ejemplo de componente React*

A su vez, dentro de esta biblioteca JS podemos utilizar otras librerías que amplíen sus capacidades tanto estéticas como funcionales. En este proyecto hemos usado las siguientes librerías de React.

### 5.5.1. MATERIAL-UI

Material UI proporciona componentes basados en la guía de Material-Design que ayudan a un desarrollo más rápido y sencillo.



*Ilustración 8. Material-UI*

### 5.5.2. RECHARTS

Recharts es otra librería de React que proporciona gráficos construidos en componentes de React.

## Recharts

A composable charting library built on React components

⚡ Install v1.8.5

🌟 Star 14,186

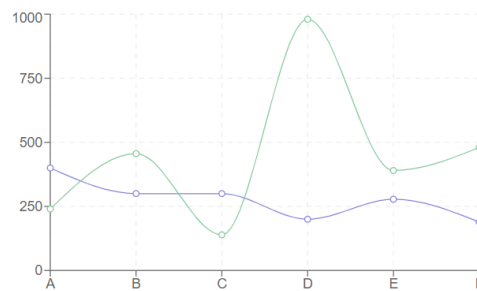


Ilustración 9. Recharts

## 5.6. XAMPP

Para poder probar y desarrollar la parte de servidor de la aplicación se ha usado la aplicación de XAMPP. XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl.[17]



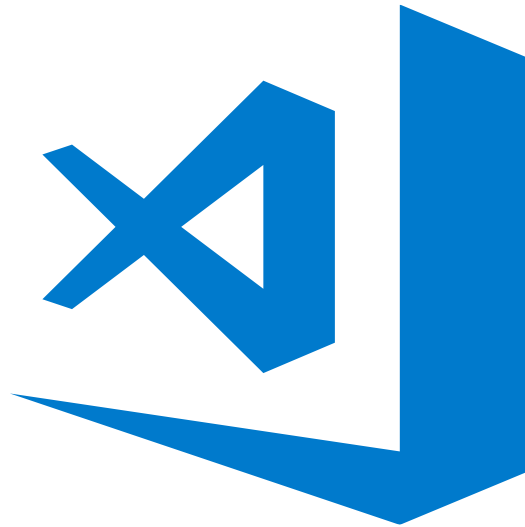
*Ilustración 10. XAMPP*

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

El programa se distribuye con la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. A esta fecha, XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.[18]

## 5.7. VISUAL STUDIO CODE

Como editor de código para el proyecto he usado Visual Studio Code. Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.[19]



*Ilustración 11. Visual Studio Code*

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. [19]

## 5.8. GITHUB

Como repositorio del proyecto hemos usado github. GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git.



*Ilustración 12. GitHub*

## 5.9. SUBLIME MERGE

Para facilitar el uso de github hemos usado el cliente de git, sublime Merge, el cual presenta una interfaz simple y sencilla de utilizar que nos permite controlar nuestro código y subirlo a github.



*Ilustración 13. Sublime Merge*

## 5.10. ADOBE XD

Como herramienta para diseñar la interfaz hemos utilizado Adobe XD, haciendo uso de las plantillas que el propio programa proporciona para el desarrollo basado en la guía de Material Design.



*Ilustración 14. Adobe XD*

## 6. ASPECTOS RELEVANTES DEL DESARROLLO

En este apartado se recogen los aspectos más interesantes del desarrollo del sistema software del proyecto.

### 6.1. ANÁLISIS DE REQUISITOS

A lo largo de la captura de requisitos, se estableció un modelo de sistema incremental basado en fases, en las que se irían añadiendo nuevas funcionalidades tras cada nueva fase. En base a esto, en el análisis de requisitos se recogen todas las funcionalidades que el sistema debería de tener estableciendo prioridades entre funcionalidades en forma de estas fases. De esta forma, las funcionalidades básicas del sistema corresponden a la fase 0, la fase que se entrega junto a esta memoria. Y las funcionalidades futuras contempladas en el análisis de requisitos, corresponden a las sucesivas fases del desarrollo.

### 6.2. ARQUITECTURA

Con el fin de lograr la mayor versatilidad y modularidad posible se optó por la división del sistema en dos partes bien diferenciadas:

- El backend: Formando la estructura lógica del proyecto, es la encargada de proporcionar el acceso a la base de datos y de las operaciones vinculadas a esta. Este acceso se permitirá mediante un sistema de servicios web tipo REST, la cual mediante URIs y los correspondientes verbos HTTP. Para crear este servicio web se decidió utilizar el framework PHP Yii2.
- El frontend: La vista, la parte del sistema con la que el usuario interactúa. Esta parte hace uso del servicio web descrito en el punto anterior para acceder a los datos alojados en la base de datos. Para crear este cliente se decidió utilizar React JS.

Esta división proporciona ventajas a largo plazo, permitiendo crear clientes en distintas plataformas pero que en su conjunto utilicen la misma lógica de negocio y acceso a datos. En nuestro caso tenemos una app web, pero se podría usar el mismo backend para una aplicación nativa, indiferentemente del sistema operativo nativo que utiliza.

### 6.3. APRENDIZAJE

Dado el sistema de arquitectura elegida de dividir el sistema en dos partes con sus respectivos lenguajes de programación, el tiempo destinado al aprendizaje de estos frameworks y lenguajes ha ocupado gran parte del tiempo dedicado a este proyecto.

Dado que para el cliente se decidió usar React JS fue necesario aprender, en primer lugar, el uso y funcionamiento del lenguaje JavaScript y posteriormente el lenguaje JSX para la creación de componentes. Dentro de React JS fue necesario comprender conceptos nuevos, como el concepto de state, props y el uso del DOM que hace React con su virtual DOM.

Para el framework Yii2 usado en el backend se utilizaron los conocimientos previos adquiridos en la asignatura “Desarrollo de Aplicaciones Web II” en las que se utilizaba el framework. Pero estos conocimientos tuvieron que ser ampliados para poder crear la API del sistema, apoyándonos en la guía que provee el propio framework.

### 6.4. DISEÑO DE LA INTERFAZ

Uno de los principales aspectos en el se ha centrado este proyecto ha sido en mantener una interfaz lo mas simple e intuitiva posible, la cual se capaz de ser usada y entendida por cualquier tipo de usuario, independientemente de su edad o educación financiera.

Con este objetivo decidimos usar la guía de diseño Material Design la cual es usada por la mayoría de las aplicaciones de Android, uno de los sistemas móviles más usados actualmente. Es por eso por lo que se eligió esta guía de diseño, la mayoría de los usuarios ya están habituados a ella y conocen la mayoría de las metáforas visuales usadas en ella.

Para mantener esta familiaridad con el usuario, los iconos usados dentro de la aplicación también siguen las líneas de Material Design, usando los iconos que proporciona esta guía de diseño.





Ilustración 15. Pantalla Principal vista de ordenador

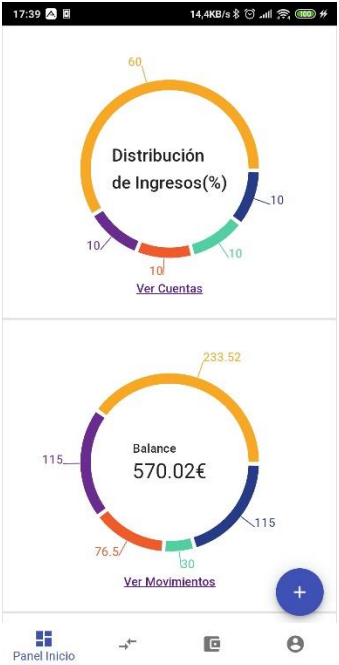


Ilustración 16. Pantalla Principal vista desde un móvil

## 7. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURO

### 7.1. CONCLUSIONES

Este proyecto surgió con el objetivo de crear una aplicación que supusiese una ayuda en la gestión de la economía personal de la gente, independientemente de su situación financiera. Este proyecto se basa en la premisa de que un ahorro planificado y controlado siempre será mejor que un ahorro ocasional basada en la situación económica actual del usuario.

Basándonos esta premisa diseñamos una aplicación centrada en la sencillez y facilidad de uso para cualquier tipo de usuario, aportándole en cada momento solo aquello que necesita, incrementando la usabilidad del sistema. Con este fin, el de la usabilidad, se optó por desarrollar un modelo de app basado en web, pudiendo acceder al sistema tanto desde un ordenador, como desde un dispositivo móvil de forma que pueda ser usado en cualquier situación.

Para lograr este sistema de app basado en web se optó por la solución de dividir el proyecto en dos partes diferenciadas, el frontend y el backend, como ya hemos comentado a lo largo de esta memoria. Esta división permite crear futuras aplicaciones o integraciones con terceros que amplíen la funcionalidad del sistema.

### 7.2. LÍNEAS DE TRABAJO FUTURAS

Como hemos comentado anteriormente, este proyecto se diseñó con el objetivo de que sus funcionalidades fueran ampliadas en un futuro. En concreto estas son las líneas de trabajo futuras:

- Permitir el agrupamiento de movimientos por categorías creadas por el propio usuario
- Permitir que las empresas que lo deseen publiquen sus productos dentro de la aplicación integrando la API del sistema en sus sistemas ERP.
- Permitir a los usuarios asociar estos productos a sus gastos.

## 8. 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>MVC</i>	Modelo Vista Controlador
<i>URI</i>	identificador de recursos uniforme ( <i>uniform resource identifier</i> )
<i>JSON</i>	notación de objeto de JavaScript ( <i>JavaScript Object Notation</i> )
<i>HTTP</i>	Protocolo de transferencia de hipertexto ( <i>Hypertext Transfer Protocol</i> )
<i>DOM</i>	Modelo de Objetos del Documento ( <i>Document Object Model</i> )
<i>JS</i>	JavaScript
<i>ERP</i>	Enterprise Resource Planning

## 9. BIBLIOGRAFÍA

- [1] F. J. García Peñalvo, J. M. Maudes Raedo, M. G. Piattini Velthuis, J. R. García-Bermejo Giner, and M. N. Moreno García, “Proyecto de Final de Carrera en la Ingeniería Técnica en Informática: Guía de Realización y Documentación,” 2000.
- [2] “Crisis económica y cómo salimos de ella. | Economía para todos.” [Online]. Available: <https://blog.iese.edu/martinezabascal/2016/11/03/crisis-y-como-salimos-de-ella-resumen-1/>. [Accessed: 12-Jun-2020].
- [3] “INE. Instituto Nacional de Estadística.” [Online]. Available: <https://www.ine.es/index.htm>. [Accessed: 12-Jun-2020].
- [4] “España - Producto Interior Bruto (PIB) de España, según la Contabilidad Nacional Trimestra.” [Online]. Available: <https://www.epdata.es/datos/pib-espana-ine-contabilidad-nacional-trimestra/36/espana/106>. [Accessed: 12-Jun-2020].
- [5] “Paro en España hoy, según la EPA del INE.” [Online]. Available: <https://www.epdata.es/datos/paro-espana-hoy-epa-ine/10/espana/106>. [Accessed: 12-Jun-2020].
- [6] C. Y. B. S. MINISTERIO DE SANIDAD, “Crisis económica y salud en España,” 2018.
- [7] Cigna, “360 Wellbeing Survey 2019: Well and Beyond’.”
- [8] “Web Services Glossary.” [Online]. Available: <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>. [Accessed: 13-Jun-2020].
- [9] “Transferencia de Estado Representacional - Wikipedia, la enciclopedia libre.” [Online]. Available: [https://es.wikipedia.org/wiki/Transferencia\\_de\\_Estado\\_Representacional](https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional). [Accessed: 13-Jun-2020].
- [10] “JSON - Wikipedia, la enciclopedia libre.” [Online]. Available: <https://es.wikipedia.org/wiki/JSON>. [Accessed: 13-Jun-2020].
- [11] “Framework - Wikipedia, la enciclopedia libre.” [Online]. Available: <https://es.wikipedia.org/wiki/Framework>. [Accessed: 13-Jun-2020].

- [12] “Apuntes Ingeniería del Software GIISI Tema 3: Modelos de proceso.”
- [13] A. Durán and T. Beatriz Bernárdez Jiménez, “Metodología para la Elicitación de Requisitos de Sistemas Software Versión 2.1.”
- [14] “Introduction - Material Design.” [Online]. Available: <https://material.io/design/introduction#goals>. [Accessed: 13-Jun-2020].
- [15] “Introducción: Acerca de Yii | Guía Definitiva de Yii 2.0 | Yii PHP Framework.” [Online]. Available: <https://www.yiiframework.com/doc/guide/2.0/es/intro-yii>. [Accessed: 13-Jun-2020].
- [16] “React - Wikipedia, la enciclopedia libre.” [Online]. Available: <https://es.wikipedia.org/wiki/React>. [Accessed: 13-Jun-2020].
- [17] “XAMPP Installers and Downloads for Apache Friends.” [Online]. Available: <https://www.apachefriends.org/es/index.html>. [Accessed: 13-Jun-2020].
- [18] “XAMPP - Wikipedia, la enciclopedia libre.” [Online]. Available: <https://es.wikipedia.org/wiki/XAMPP>. [Accessed: 13-Jun-2020].
- [19] “Visual Studio Code - Wikipedia, la enciclopedia libre.” [Online]. Available: [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code). [Accessed: 13-Jun-2020].