



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
G^o en Ingeniería en Informática



TFG Ingeniería Informática:

**Análisis comercial urbano
utilizando series temporales de
compras de tarjetas de crédito**



Presentado por Sergio Bueno Medina
en Burgos el 13 de febrero de 2020

Tutores Dr. José Manuel Galán Ordax y Virginia
Ahedo García

Índice General

Índice General	1
Índice de figuras	2
Índice de tablas	3
Apéndice A. Planificación	4
A.1. Introducción	4
A.2. Planificación temporal.....	5
A.3. Estudio económico	13
A.4. Tratamiento de licencias.....	15
Apéndice B. Especificación de Requisitos.....	17
B.1. Introducción	17
B.2. Objetivos generales.....	17
B.3. Catálogo de requisitos	18
B.4. Especificación de requisitos	19
Apéndice C. Especificación de Diseño	29
C.1. Diseño de datos.....	29
C.2. Diseño arquitectónico	33
C.3. Diseño de interfaces.....	33
Apéndice D. Documentación Técnica de Programación	37
D.1. Introducción	37
D.2. Estructura de directorios	37
D.3. Manual del programador.....	38
Apéndice E. Documentación de Usuario	42
E.1. Introducción	42
E.2. Requisitos de usuarios.....	42
E.3. Instalación	42
E.4. Manual de Usuario.....	42

Índice de figuras

Figura A 1 Gráfico burndown Sprint 1.....	5
Figura A 2 Gráfico burndown Sprint 2.....	7
Figura A 3 Gráfico burndown Sprint 3.....	8
Figura A 4 Gráfico burndown Sprint 4.....	9
Figura A 5 Gráfico burndown Sprint 5.....	10
Figura A 6 Gráfico burndown Sprint 6.....	11
Figura A 7 Gráfico burndown Sprint 7.....	12
Figura A 8 Gráfico burndown Sprint 8.....	¡Error! Marcador no definido.

Figura C 1 Diseño de datos	32
Figura C 2 Relaciones	33
Figura C 3 Diseño arquitectónico.....	33
Figura C 4 Prototipo página de acceso.....	34
Figura C 5 Prototipo registro.....	34
Figura C 6 Prototipo página principal.....	34
Figura C 7 Prototipo gráficos.....	35
Figura C 8 Prototipo tablas.....	35
Figura C 9 Prototipo tabla con todos los datos.....	36
Figura C 10 Prototipo menú administración.....	36

Figura E 1 Página de acceso	43
Figura E 2 Página de registro	43
Figura E 3 Cambio de idioma.....	44
Figura E 4 Menú con rol de administrador	44
Figura E 5 Menú con rol de usuario	44
Figura E 6 Menú con rol de usuario	45
Figura E 7 Página de inicio	46
Figura E 8 Página de administración.....	46
Figura E 9 Datos básicos.....	47
Figura E 10 Datos por categoría.....	47
Figura E 11 Datos por día	48
Figura E 12 Datos por hora	48
Figura E 13 Datos por destino.....	49
Figura E 14 Datos por origen.....	49
Figura E 15 Datos por origen y edad	50
Figura E 16 Datos por origen y género	50
Figura E 17 Mostrar tablas	50
Figura E 18 Mostrar todos los datos.....	50
Figura E 19 Descargar fichero .net.....	51

Figura E 20 Tablas	51
Figura E 21 Descargar como csv	52
Figura E 22 Salir de la aplicación.....	52
Figura E 23 Atrás.....	52

Índice de tablas

Tabla A. 1 Story points del proyecto.....	4
Tabla A. 2 Etiquetas Sprint 1	6
Tabla A. 3 Etiquetas Sprint 2	7
Tabla A. 4 Etiquetas Sprint 3	8
Tabla A. 5 Etiquetas Sprint 4	9
Tabla A. 6 Etiquetas Sprint 5	11
Tabla A. 7 Etiquetas Sprint 6	12
Tabla A. 8 Etiquetas Sprint 7	13
Tabla A. 9 Etiquetas Sprint 8	¡Error! Marcador no definido.
Tabla A. 10 Licencias de dependencias	16
Tabla B. 1 Registro de usuario	20
Tabla B. 2 Acceso de un usuario.....	21
Tabla B. 3 Mostar gráficos	21
Tabla B. 4 Mostrar tablas	22
Tabla B. 5 Mostrar tabla con todos los datos	22
Tabla B. 6 Descargar tabla como csv	23
Tabla B. 7 Internacionalización	23
Tabla B. 8 Salir.....	24
Tabla B. 9 Seleccionar código postal.....	24
Tabla B. 10 Seleccionar mes.....	25
Tabla B. 11 Seleccionar día.....	25
Tabla B. 12 Atrás	26
Tabla B. 13 Obtener posiciones personajes	26
Tabla B. 14 Extraer datos.....	27
Tabla B. 15 Subir datos	27
Tabla B. 16 Extraer y subir datos.....	28
Tabla B. 17 Acceder a menú de administración	28

Apéndice A. Planificación

A.1. Introducción

Para el desarrollo del proyecto, como ya hemos mencionado en la memoria, se ha empleado la metodología Scrum por lo que para cada tarea se ha definido una etiqueta y una estimación temporal del tiempo requerido para realizarla.

Para poder realizar está metodología se ha empleado la herramienta ZenHub.

Las etiquetas disponibles para cada tarea son:

- Desarrollo inicial: Si se realizó durante el desarrollo inicial del proyecto.
- Mejoras: Si la tarea realiza una mejora sobre la aplicación.
- Epic: Si la tarea es un Epic compuesto de otras tareas.
- Tablas: Si la tarea implementa algo relacionado con tablas.
- Gráficos: Si la tarea realiza una mejora relacionada con gráficos

Para cada estimación se emplea Story Points que se corresponden al tiempo necesario para realizar una tarea. Se Definen con la siguiente tabla:

Story Points	Estimación temporal
1	1 hora
2	2 horas
3	3 horas
5	5 horas
8	8 horas

Tabla A. 1 Story points del proyecto

A.2. Planificación temporal

En este apartado defino los sprint realizados durante el proyecto:

Sprint 1.

Configuración inicial Symfony.

Se inicia el proyecto y se configura de forma inicial el Framework Symfony.

Link : [Sprint 1](#)

Fecha inicio / fin : 23 octubre 2019 / 26 octubre 2019

Objetivo : Iniciar el proyecto y realizar la configuración inicial del entorno de desarrollo

Etiquetas : Desarrollo Inicial, Epic

Story Points : 19

Gráfico Burndown.



Figura A 1 Gráfico burndown Sprint 1

Etiquetas.

Issues	Titulo	Desarrollo Inicial	Epic	Gráficos	Mejoras	Tablas
#1	Problemas iniciales con ZenHub					
#2	Instalación inicial symfony					
#3	Instalación Apache2					
#4	Instalación PHP					
#5	Instalar Composer					
#6	Instalar NodeJs y Npm					
#7	Configuración inicial Symfony					
#8	Instalación de mysql en local como base de datos					

Tabla A. 2 Etiquetas Sprint 1

Sprint 2.

Creación de usuarios y seguridad de la aplicación.

Se quiere añadir a la aplicación control de acceso por usuarios con posible registro y página de acceso.

Link : [Sprint 2](#)

Fecha inicio / fin : 26 octubre 2019 / 4 noviembre 2019

Objetivo : Añadir página de registro y página de acceso

Etiquetas : Desarrollo Inicial

Story Points : 13

Gráfico burndown.



Figura A 2 Gráfico burndown Sprint 2

Etiquetas.

Issues	Titulo	Desarrollo Inicial	Epic	Gráficos	Mejoras	Tablas
#9	Login de la página					
#10	Registro de la página					

Tabla A. 3 Etiquetas Sprint 2

Sprint 3.

Recoger datos de la API y meterlos en base de datos local.

Se comienza a recoger los datos de la API de BBVA e integrarlo en nuestra base de datos.

Link : [Sprint 3](#)

Fecha inicio / fin : 4 noviembre 2019 / 24 noviembre 2019

Objetivo : Recoger datos de la API y guardarlos en base de datos

Etiquetas : Desarrollo Inicial

Story Points : 16

Gráfico burndown.



Figura A 3 Gráfico burndown Sprint 3

Etiquetas.

Issues	Titulo	Desarrollo Inicial	Epic	Gráficos	Mejoras	Tablas
#11	Categorías y subcategorías de tipos de mercados					
#12	Dividir el App Controller en diferentes funciones					
#13	Integrar Sass usando Gulp para convertirlo a css					
#14	Recoger datos basic_stats de la API BBVA y pasarlos a base de datos local					

Tabla A. 4 Etiquetas Sprint 3

Sprint 4.

Cambiar método de guardar peticiones de la API en base de datos.

Debido al tamaño de los datos recibidos, mayores de 500000 registros en algunos casos, necesito optimizar la forma de subir datos en mi base de datos tanto en memoria como en tiempo

Link : [Sprint 4](#)

Fecha inicio / fin : 2 diciembre 2019 / 6 diciembre 2019

Objetivo : Cambiar forma de subir datos debido a gran cantidad de registros en algunos casos

Etiquetas : Desarrollo Inicial

Story Points : 13

Gráfico burndown.

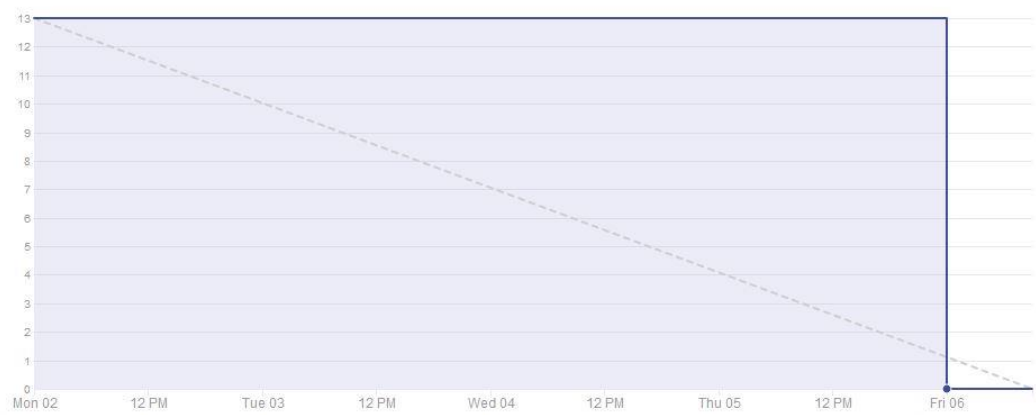


Figura A 4 Gráfico burndown Sprint 4

Etiquetas.

		Desarrollo Inicial	Epic	Gráficos	Mejoras	Tablas
Issues	Titulo					
#15	Cambiar category data					
#16	Cambiar day data y hour data					
#17	Añadir datos distribución por destino					
#18	Añadir datos de distribución por origen					

Tabla A. 5 Etiquetas Sprint 4

Sprint 5.

Implementar diseño de la página.

Implementar el diseño inicial de la página, así como los iconos necesarios.

Link : [Sprint 5](#)

Fecha inicio / fin : 21 diciembre 2019 / 4 enero 2019

Objetivo : Implementar diseño inicial de la aplicación.

Etiquetas : Mejoras

Story Points : 18

Gráfico burndown.

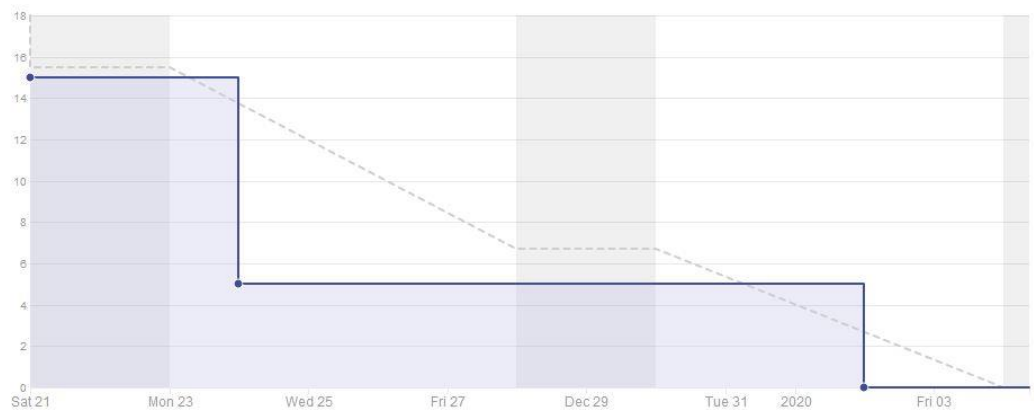


Figura A 5 Gráfico burndown Sprint 5

Etiquetas.

Issues	Titulo	Desarrollo Inicial	Epic	Gráficos	Mejoras	Tablas
#19	Implementar diseño ventana de <i>login</i>					
#20	Implementar diseño página de registro					
#21	Implementar diseño página principal					
#22	Añadir datos de distribución por origen					

Tabla A. 6 Etiquetas Sprint 5

Sprint 6.

Agregar Tablas sobre aplicación web.

Agregar tablas sobre mi aplicación.

Link : [Sprint 6](#)

Fecha inicio / fin : 8 enero 2019 / 15 enero 2019

Objetivo : Implementar las tablas

Etiquetas : Tablas

Story Points : 24

Gráfico burndown.



Figura A 6 Gráfico burndown Sprint 6

Etiquetas.

Issues	Titulo	Desarrollo Inicial	Epic	Gráficos	Mejoras	Tablas
#23	Implementar tabla datos básicos					
#24	Implementar tabla datos por categoría					

#25	Implementar tabla datos por horas	
#26	Implementar tabla datos por origen	
#27	Implementar tabla datos por destino	
#28	Implementar tabla datos por origen y edad	
#29	Implementar tabla de origen por género	
#30	Implementar tabla por día	

Tabla A. 7 Etiquetas Sprint 6

Sprint 7.

Agregar Gráficos sobre aplicación web.

Link : [Sprint 7](#)

Fecha inicio / fin : 15 enero 2019 / 22 enero 2019

Objetivo : Implementar gráficos

Etiquetas : Gráficos

Story Points : 18

Gráfico burndown.

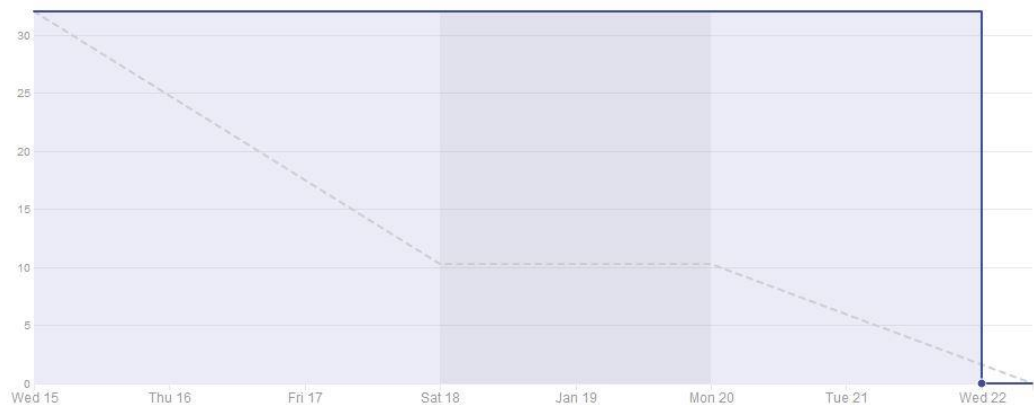


Figura A 7 Gráfico burndown Sprint 7

Etiquetas.

		Desarrollo Inicial	Epic	Gráficos	Mejoras	Tablas
Issues	Titulo					
#31	Implementar gráficos datos básicos					
#32	Implementar gráficos datos por categoría					
#33	Implementar gráficos por origen					
#34	Implementar gráficos por origen y edad					
#35	Implementar gráficos por destino					
#36	Implementar gráficos por origen y género					
#37	Implementar gráficos por día					
#38	Implementar gráficos por hora					

Tabla A. 8 Etiquetas Sprint 7

Sprint 8.

Continúo con la documentación y mejoras finales y refactorización de la aplicación.

Link : [Sprint 8](#)

Fecha inicio / fin : 25 enero 2020 / 13 febrero 2020

Objetivo : Cambios finales de la aplicación

Etiquetas : Mejoras

Story Points : 32

Gráfico burndown.

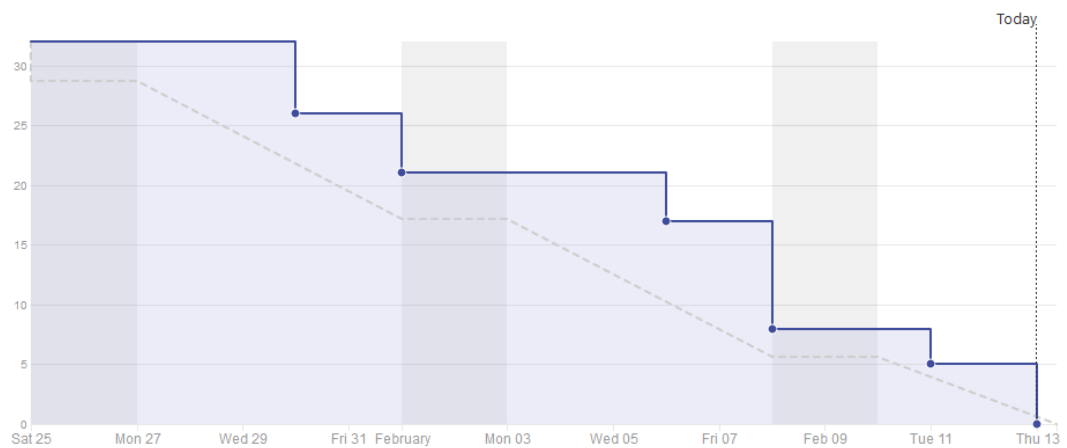


Figura A 8 Gráfico burndown Sprint 8

Etiquetas.

		Desarrollo Inicial	Epic	Gráficos	Mejoras	Tablas
Issues	Titulo					
#39	Añado selector por año					
#40	Refactorizar la aplicación					
#41	Añadir traductor por año					
#42	Añado e implemento idiomas					
#43	Mejoras para la versión de móvil					
#44	Solucionar error al elegir mes sin datos					
#45	Solucionar problema al recoger datos de la API					
#46	Despliegue de la aplicación en servidor VPS en la nube					
#47	Implementar opcion de descargar datos e destino y origen como un fichero con extension .net					

Tabla A. 9 Etiquetas Sprint 8

A.3. Estudio económico

Realizo un estudio del coste real de implementar el proyecto de forma completa.

Viabilidad económica.

Para que el proyecto genere ingresos se podría gestionar un método de suscripción para emplear el servicio, pero en principio el proyecto solo se basa en el ámbito docente puesto que haría falta comprar licencia para hacer uso de la API BBVA.

Costes software.

Todas las herramientas utilizadas son de uso gratuito y licencias open Source por lo que no hay coste por esta parte.

Costes hardware.

No existe ningún coste asociado al hardware para la aplicación, aunque dependerá dónde se aloje la aplicación final.

Costes de personal.

Se considera que el proyecto se ha realizado por un programador durante 6 horas diarias, durante 4 meses:

Concepto	Coste
Salario mensual neto	1000 €
Formación profesional	166,67 €
Tasa por desempleo (5,5%)	64,17 €
Contingencias comunes (23,6%)	275,33 €
Fogasa (0,2%)	2,33 €
Formación profesional (0,6%)	7 €
Coste mensual	1515,5 €
TOTAL	6062 €

A.4. Tratamiento de licencias

Para cada una de las herramientas utilizadas hemos empleado licencias libres de uso y comercialización. La única restricción es que se deben incluir las licencias en todas las copias o partes sustanciales del código.

Módulo	Versión	Licencia
Symfony 4	4.4	MIT
Git	2.17.1	GNU GPL v2
Zotero	5.0	GNU AGPL v3
Pencil Project	3.1.0	GNU GPL v2
MySQL	14.14	GNU GPL v2
Npm	6.13.6	ISC
Node JS	12.14.1	MIT
Composer	1.9.0	MIT

Tabla A. 10 Licencias de dependencias

La licencia empleada en el proyecto es MIT (*Massachusetts Institute Technology*), ya que es una licencia de uso libre y no requiere ninguna garantía de funcionamiento o mal uso de la misma.

No se incluyen los paquetes de npm ni de los de composer ya que se encuentran incluidos dentro de los gestores de paquetes.

En caso de ver el listado completo se pueden ver las licencias en los ficheros composer.lock y package.json dentro de la carpeta /src

Apéndice B. Especificación de Requisitos

B.1. Introducción

Descripción de los requisitos funcionales y no funcionales de la aplicación WEB.

B.2. Objetivos generales

Los objetivos generales de la aplicación son:

- Desarrollar una aplicación WEB con Symfony.
- Permitir el registro de nuevos usuarios mediante una página de registro.
- Implementar acceso a la misma mediante página de acceso.
- Controlar el acceso a la aplicación según el rol del usuario.
- Permitir extraer datos de la API BBVA y subir los datos obtenidos a nuestra base de datos.
- Mostrar tablas asociadas a los datos obtenidos por cada código postal.
- Permitir descargar los datos como csv.
- Mostrar gráficos asociados a los datos obtenidos por cada código postal.

B.3. Catálogo de requisitos

Requisitos funcionales

- **RF-1 Registro de un nuevo usuario.** Un usuario se registra en la aplicación.
- **RF-2 Validación de datos introducidos por el usuario.** Se comprueban los datos introducidos por el usuario dentro de un formulario.
- **RF-3 Usuario accede con un usuario previamente registrado.** Usuario accede a la aplicación con un nombre de usuario determinado.
- **RF-4 Mostrar gráficos con datos.** Mostrar gráficos con cada uno de los datos.
- **RF-5 Mostrar tablas con datos.** Mostrar tablas con cada uno de los datos.
- **RF-6 Mostrar todos los datos.** Obtener tabla con todos los datos disponibles en base de datos.
- **RF-7 Descargar tabla como csv.** Descargar tabla actual como csv.
- **RF-8 Internacionalización.** Se puede cambiar el idioma actual.
- **RF-9 Salir.** Salir de la aplicación a la página de acceso.
- **RF-10 Seleccionar código postal.** Cambiar código postal desde el que obtener datos.
- **RF-11 Seleccionar mes.** Seleccionar mes del que obtener datos.
- **RF-12 Seleccionar día.** Seleccionar día del que obtener datos.
- **RF-13 Atrás.** Botón desde el que volver a la página anterior.
- **RF-14 Página principal.** Al pulsar sobre el título se vuelve a la página principal.
- **RF-15 Acceder a menú de administración.** Se muestran los botones para hacer peticiones a la API y poder descargar los datos.

- **RF-16 Extraer datos.** Extraer datos de la API.
- **RF-17 Subir datos.** Subir datos extraídos a la base de datos.

Requisitos no funcionales

- **RNF-1 Seguridad:** Se garantiza la seguridad de la aplicación y el uso de los datos al no almacenar sin encriptar ningún dato del usuario.
- **RNF-2 Rendimiento:** Se emplea como medida de rendimiento, el tiempo dado que es una aplicación web.
- **RNF-3 Modularidad:** La aplicación permite la extensibilidad de la aplicación al poder integrar más módulos.
- **RNF-4 Internacionalización:** La aplicación dispone de varios idiomas.

B.4. Especificación de requisitos

Actores:

- **Usuario no identificado:** Puede acceder a la página de registro y de acceso.
- **Usuario identificado:** Puede utilizar la aplicación y acceder a diferentes partes de la misma según el rol de usuario.

CU - 01	Registro de un nuevo usuario	
Descripción	El usuario se registra en la aplicación web	
Precondiciones	El usuario accede a la página de registro	
Secuencia Normal	Paso	Acción
	1	El usuario accede a la página de registro
	2	El usuario introduce sus datos
	3	El usuario envía los datos
Excepciones	Paso	Acción
	3	Tras validar los datos se comprueba que no son correctos
	3	Se produce una excepción al almacenar los datos en base de datos
Postcondiciones	Se almacena el nuevo usuario en base de datos	

Frecuencia esperada	Baja
Importancia	Alta
Comentarios	Cumple RF-1, RF-2

Tabla B. 1 Registro de usuario

CU - 02	Acceso de un usuario	
Descripción	Un usuario accede a la aplicación introduciendo su usuario y contraseña.	
Precondiciones	El usuario se ha registrado previamente	
Secuencia Normal	Paso	Acción
	1	El usuario accede a la página de acceso
	2	El usuario introduce sus datos
	3	El usuario envía los datos
Excepciones	Paso	Acción
	3	Tras validar los datos se comprueba que no son correctos
	3	Se produce una excepción al almacenar los datos en base de datos
Postcondiciones	Se accede a la aplicación	
Frecuencia esperada	Media	
Importancia	Alta	
Comentarios	Cumple RF-3, RF-2	

Tabla B. 2 Acceso de un usuario

CU - 03	Mostrar gráficos	
Descripción	Para cada uno de los tipos de datos se accede a los gráficos asociados.	
Precondiciones	El usuario ha accedido a la aplicación	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre uno de los botones de datos del menú principal
Excepciones	Paso	Acción
	1	Se produce una excepción al cargar los datos
Postcondiciones	Se muestran los gráficos asociados al tipo de dato seleccionado	
Frecuencia esperada	Alta	
Importancia	Media	
Comentarios	Cumple RF-4	

Tabla B. 3 Mostar gráficos

CU - 04	Mostrar tablas	
Descripción	Una vez que se accede al listado de gráficos se puede mostrar la tabla con los datos asociados a los gráficos.	
Precondiciones	Acceder previamente a los gráficos de cualquier tipo de dato	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre el botón “Mostrar tabla asociada a los datos”
Excepciones	Paso	Acción
	1	Se produce una excepción al mostrar la tabla
Postcondiciones	Se muestra la tabla seleccionada	
Frecuencia esperada	Alta	
Importancia	Media	
Comentarios	Cumple RF-5	

Tabla B. 4 Mostrar tablas

CU - 05	Mostrar todos los datos	
Descripción	Mostrar tabla con datos de todos los códigos postales disponibles.	
Precondiciones	Acceder previamente a cualquier tabla asociada a los gráficos.	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre el botón “Mostrar todos los datos”
Excepciones	Paso	Acción
	1	Se produce una excepción al mostrar la tabla
Postcondiciones	Se muestra la tabla seleccionada con datos de todos los códigos postales.	
Frecuencia esperada	Baja	
Importancia	Media	
Comentarios	Cumple RF-6	

Tabla B. 5 Mostrar tabla con todos los datos

CU - 06	Descargar tabla como csv	
Descripción	Se descarga la tabla actual como csv	
Precondiciones	Haber accedido a cualquier tabla	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre el botón “Descargar tabla como fichero csv”
Excepciones	Paso	Acción
Postcondiciones	Se obtiene el fichero csv asociado a la tabla	
Frecuencia esperada	Baja	
Importancia	Baja	
Comentarios	Cumple RF-7	

Tabla B. 6 Descargar tabla como csv

CU - 07	Internacionalización	
Descripción	Se permite cambiar el idioma de la aplicación en cualquier ventana de la aplicación	
Precondiciones		
Secuencia Normal	Paso	Acción
	1	Desde cualquier página de la aplicación, incluso sin haber accedido se puede elegir el idioma en el desplegable de la parte superior derecha.
Excepciones	Paso	Acción
Postcondiciones	Se cambia el lenguaje empleado por la aplicación	
Frecuencia esperada	Baja	
Importancia	Alta	
Comentarios	Cumple RF-8	

Tabla B. 7 Internacionalización

CU - 08	Salir	
Descripción	Un usuario sale de la aplicación	
Precondiciones	Haber accedido a la aplicación	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre el botón “Salir” en la parte superior derecha de la aplicación
Excepciones	Paso	Acción
Postcondiciones	Se sale de la aplicación volviendo a la página de acceso inicial	
Frecuencia esperada	Media	
Importancia	Media	
Comentarios	Cumple RF-9	

Tabla B. 8 Salir

CU - 09	Seleccionar código postal	
Descripción	Un usuario cambia el código postal actual.	
Precondiciones	Estar dentro de una página de la aplicación donde es posible cambiar el código postal sobre los datos actuales	
Secuencia Normal	Paso	Acción
	1	Elegir código postal en el desplegable
Excepciones	Paso	Acción
Postcondiciones	Se cambian los datos mostrados con respecto al código postal seleccionado	
Frecuencia esperada	Media	
Importancia	Media	
Comentarios	Cumple RF-10	

Tabla B. 9 Seleccionar código postal

CU - 10	Seleccionar mes	
Descripción	Un usuario cambia el mes actual.	
Precondiciones	Estar dentro de una página de la aplicación donde es posible cambiar el mes sobre los datos actuales	
Secuencia Normal	Paso	Acción
	1	Elegir mes en el desplegable
Excepciones	Paso	Acción
Postcondiciones	Se cambian los datos mostrados con respecto al mes seleccionado	
Frecuencia esperada	Media	
Importancia	Media	
Comentarios	Cumple RF-11	

Tabla B. 10 Seleccionar mes

CU - 11	Seleccionar día	
Descripción	Un usuario cambia el día actual.	
Precondiciones	Estar dentro de una página de la aplicación donde es posible cambiar el día sobre los datos actuales	
Secuencia Normal	Paso	Acción
	1	Elegir día en el desplegable
Excepciones	Paso	Acción
Postcondiciones	Se cambian los datos mostrados con respecto al día seleccionado	
Frecuencia esperada	Media	
Importancia	Media	
Comentarios	Cumple RF-12	

Tabla B. 11 Seleccionar día

CU - 12	Atrás	
Descripción	Un usuario puede volver a la página anterior	
Precondiciones	Haber accedido a la aplicación con un usuario	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre la flecha hacia atrás para volver a la página anterior
Excepciones	Paso	Acción
Postcondiciones	Se vuelve a la página anterior	
Frecuencia esperada	Media	
Importancia	Baja	
Comentarios	Cumple RF-13	

Tabla B. 12 Atrás

CU - 13	Página principal	
Descripción	Un usuario puede volver a la página principal	
Precondiciones	Haber accedido a la aplicación con un usuario	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre el título de la aplicación
Excepciones	Paso	Acción
Postcondiciones	Se vuelve a la página principal	
Frecuencia esperada	Alta	
Importancia	Alta	
Comentarios	Cumple RF-14	

Tabla B. 13 Obtener posiciones personajes

CU - 14	Extraer datos	
Descripción	Extraer datos del tipo de datos seleccionado de la API	
Precondiciones	Haber accedido al menú de administración	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre cualquiera de los botones de extraer datos.
Excepciones	Paso	Acción
	1	Se produce una excepción al extraer datos de la API.
Postcondiciones	Se extraen los datos seleccionados y se guardan en un fichero csv para posteriormente subirlos.	
Frecuencia esperada	Baja	
Importancia	Alta	
Comentarios	Cumple RF-16	

Tabla B. 14 Extraer datos

CU - 15	Subir datos	
Descripción	Subir datos desde fichero csv	
Precondiciones	Haber accedido al menú de administración	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre cualquiera de los botones de subir datos.
Excepciones	Paso	Acción
	1	Se produce una excepción al subir datos desde el fichero csv
Postcondiciones	Se suben los datos desde el fichero csv	
Frecuencia esperada	Baja	
Importancia	Alta	
Comentarios	Cumple RF-17	

Tabla B. 15 Subir datos

CU - 16	Extraer y subir datos	
Descripción	Un usuario extrae y sube los datos de mercaderes o datos básicos	
Precondiciones	Haber accedido al menú de administración	
Secuencia Normal	Paso	Acción
	1	Pulsar sobre “extraer y subir mercaderes” o “extraer y subir datos básicos”
Excepciones	Paso	Acción
	1	Se produce una excepción al extraer datos desde la API
	2	Se produce una excepción al subir datos desde fichero csv
Postcondiciones	Se extraen y suben los datos seleccionados	
Frecuencia esperada	Baja	
Importancia	Alta	
Comentarios	Cumple RF-16, RF-17	

Tabla B. 16 Extraer y subir datos

CU - 17	Acceder a menú de administración	
Descripción	Un usuario accede al menú de administración para extraer y subir datos a la base de datos	
Precondiciones	Haber accedido a la aplicación como usuario administrador.	
Secuencia Normal	Paso	Acción
	1	El usuario pulsa sobre “Extraer y subir a base de datos”
Excepciones	Paso	Acción
	1	Se produce una excepción al extraer o subir los datos
Postcondiciones	Se muestra la página de administración	
Frecuencia esperada	Baja	
Importancia	Alta	
Comentarios	Cumple RF-15	

Tabla B. 17 Acceder a menú de administración

Apéndice C. Especificación de Diseño

C.1. Diseño de datos

Para el diseño de la base de datos se ha basado en los datos recibidos desde la API. La aplicación contiene las siguientes entidades:

- Datos básicos (*basic_data*), contiene los siguientes campos:
 - *id*: Identificador de la tabla
 - *avg*: Media de gasto por uso de tarjeta por código postal
 - *cards*: Número de tarjetas
 - *date*: Fecha de los datos a mostrar
 - *txs*: Número de transacciones
 - *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
 - *min*: Gasto mínimo
 - *peak_txs_day*: Día en el que se ha realizado el mayor gasto
 - *peak_txs_hour*: Hora en la que se ha realizado el mayor gasto
 - *std*: Desviación estándar
 - *valley_txs_day*: Día en el que se realiza el menor gasto
 - *valley_txs_hour*: Hora en la que se realiza el menor gasto
 - *max*: Gasto máximo
 - *zipcode*: Código postal sobre el que se muestran los datos
- Categoría (*category*), contiene los siguientes campos:
 - *id*: Identificador de la tabla
 - *code*: Código de la categoría
 - *description*: Descripción de la categoría
 - *subCategories*: Listado de subcategorías que pertenecen a esta
 - *categoryData*: Listado de datos por categoría
- Subcategoría (*sub_category*), contiene los siguientes campos:
 - *id*: Identificador de la tabla
 - *code*: Código de la categoría
 - *description*: Descripción de la categoría
 - *category*: Categoría a la que pertenece
- Datos por categoría (*category_data*), contiene los campos:
 - *id*: Identificador de la tabla
 - *avg*: Media de gasto por uso de tarjeta por código postal
 - *cards*: Número de tarjetas
 - *date*: Fecha de los datos a mostrar
 - *txs*: Número de transacciones

- *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
 - *zipcode*: Código postal sobre el que se muestran los datos
 - *category*: Categoría a la que pertenecen los datos
- Datos por día (*day_data*), contiene los campos:
 - *id*: Identificador de la tabla
 - *avg*: Media de gasto por uso de tarjeta por código postal
 - *day*: Día sobre el que se muestran los datos
 - *cards*: Número de tarjetas
 - *date*: Fecha de los datos a mostrar
 - *txs*: Número de transacciones
 - *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
 - *max*: Gasto máximo
 - *min*: Gasto mínimo
 - *std*: Desviación estándar
 - *mode*: Moda
 - *hourData*: Listado de datos por hora
- Destino (*destination*), listado de uso de tarjeta de usuarios de un código postal determinado en otros códigos postales de destino, contiene los campos:
 - *id*: Identificador de la tabla
 - *avg*: Media de gasto por uso de tarjeta por código postal
 - *cards*: Número de tarjetas
 - *date*: Fecha de los datos a mostrar
 - *txs*: Número de transacciones
 - *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
 - *zipcode*: Código postal sobre el que se muestran los datos
 - *destinationData*: Listado con los datos por cada código postal de destino
- Datos por destino (*destination_data*), listado de destinos por cada código postal, contiene los campos:
 - *id*: Identificador de la tabla
 - *avg*: Media de gasto por uso de tarjeta por código postal
 - *cards*: Número de tarjetas
 - *destination*: Destino asociado
 - *txs*: Número de transacciones
 - *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
 - *destinationZipcode*: Código postal de destino
- Datos por hora (*hour_data*), contiene los campos:

- *id*: Identificador de la tabla
- *avg*: Media de gasto por uso de tarjeta por código postal
- *hour*: Hora sobre el que se muestran los datos
- *cards*: Número de tarjetas
- *date*: Fecha de los datos a mostrar
- *txs*: Número de transacciones
- *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
- *max*: Gasto máximo
- *min*: Gasto mínimo
- *std*: Desviación estándar
- *mode*: Moda
- *dayData*: Listado de datos por día
- Datos por origen (*origin_data*), listado de orígenes de uso de tarjeta de un código postal determinado, contiene los campos:
 - *id*: Identificador de la tabla
 - *avg*: Media de gasto por uso de tarjeta por código postal
 - *cards*: Número de tarjetas
 - *txs*: Número de transacciones
 - *date*: Fecha de los datos a mostrar
 - *zipcode*: Código postal sobre el que se muestran los datos
 - *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
 - *originZipcode*: Código postal de origen
- Datos por origen y edad (*origin_age_data*), listado de orígenes por edad, contiene los campos:
 - *id*: Identificador de la tabla
 - *avg*: Media de gasto por uso de tarjeta por código postal
 - *cards*: Número de tarjetas
 - *age*: Edad de los usuarios que usan tarjeta
 - *txs*: Número de transacciones
 - *date*: Fecha de los datos a mostrar
 - *zipcode*: Código postal sobre el que se muestran los datos
 - *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
 - *originZipcode*: Código postal de origen
- Datos por origen y género (*origin_gender_data*), listado de orígenes por género, contiene los campos:
 - *id*: Identificador de la tabla
 - *avg*: Media de gasto por uso de tarjeta por código postal
 - *cards*: Número de tarjetas
 - *gender*: Género de los usuarios que usan tarjeta

- *txs*: Número de transacciones
- *merchants*: Número de mercaderes sobre los que se ha realizado pago con tarjeta
- *originAgeData*: Listado de datos por edad
- Usuario (*user*), usuarios registrados en la página, contiene los campos:
 - *id*: Identificador de la tabla
 - *email*: Email del usuario
 - *roles*: Roles del usuario
 - *password*: Contraseña del usuario encriptada
- Código postal (*zipcode*), contiene los campos:
 - *id*: Identificador de la tabla
 - *zipcode*: Código postal
 - *basicData*: Listado de datos básicos
 - *categoryData*: Listado de datos por categoría
 - *dayData*: Listado de datos por día
 - *destinations*: Listado de destinos
 - *originData*: Listado de datos por origen
 - *originAgeData*: Listado de datos por edad

El diagrama de clases, de la base de datos, vinculado a las entidades que se acaban de describir es:

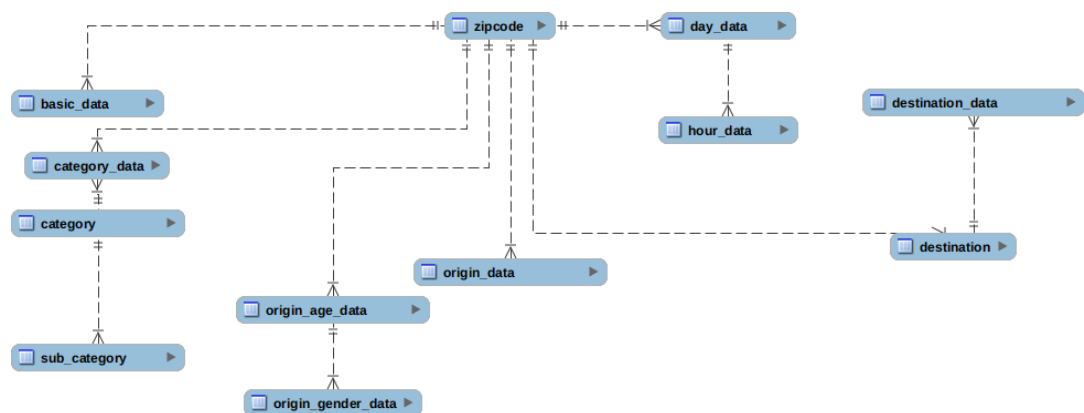


Figura C 1 Diseño de datos

Las relaciones son:

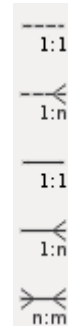


Figura C 2 Relaciones

C.2. Diseño arquitectónico

Como ya se mencionó en la memoria el modelo seguido es el Modelo Vista Controlador, éste se ha seguido durante el desarrollo de toda la aplicación. Por un lado tenemos todas las clases en la carpeta src/Entity que representa el Modelo, las clases en la carpeta src/templates que representan la Vista y por último las clases en la carpeta src/Controller que representa la parte Controlador. El diagrama que representa estas relaciones es el siguiente:

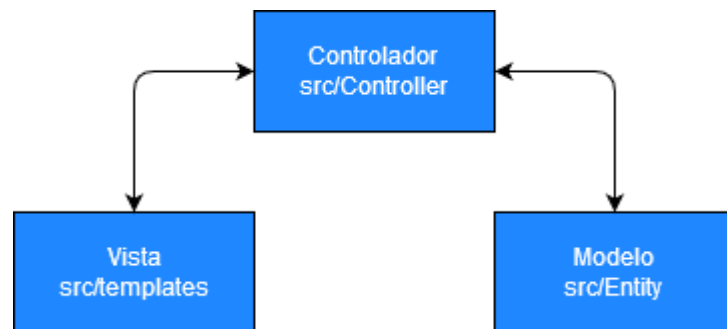


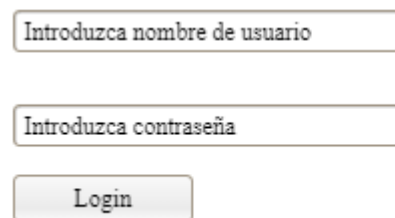
Figura C 3 Diseño arquitectónico

C.3. Diseño de interfaces

Para el diseño de prototipos se ha empleado la aplicación Pencil.

Los prototipos de la interfaz son los siguientes:

Analisis Comercial Urbano



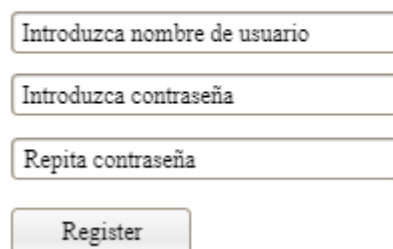
Introduzca nombre de usuario

Introduzca contraseña

Login

Figura C 4 Prototipo página de acceso

Analisis Comercial Urbano



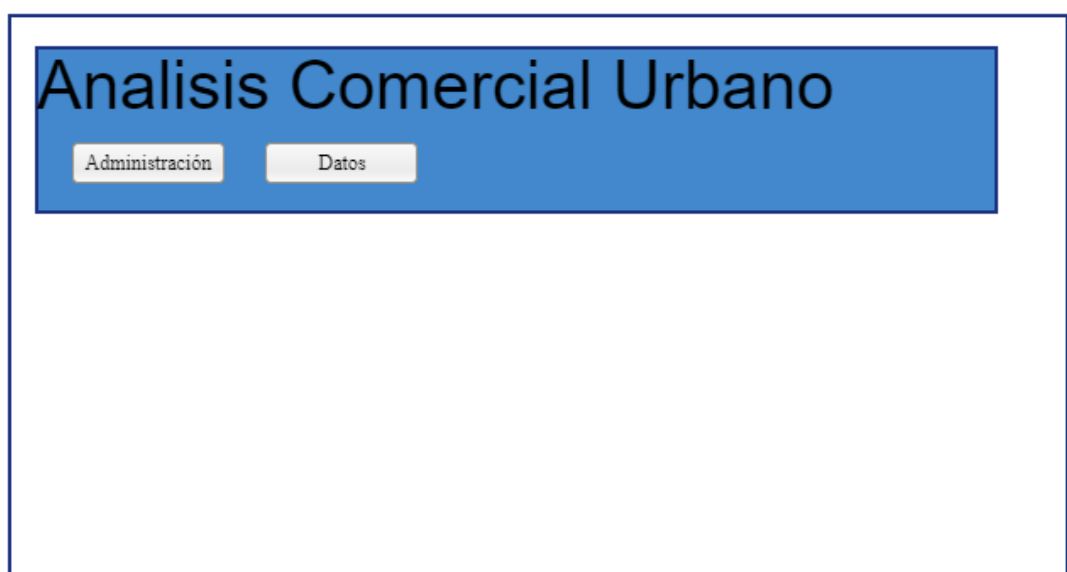
Introduzca nombre de usuario

Introduzca contraseña

Repita contraseña

Register

Figura C 5 Prototipo registro



Analisis Comercial Urbano

Administración Datos

Figura C 6 Prototipo página principal

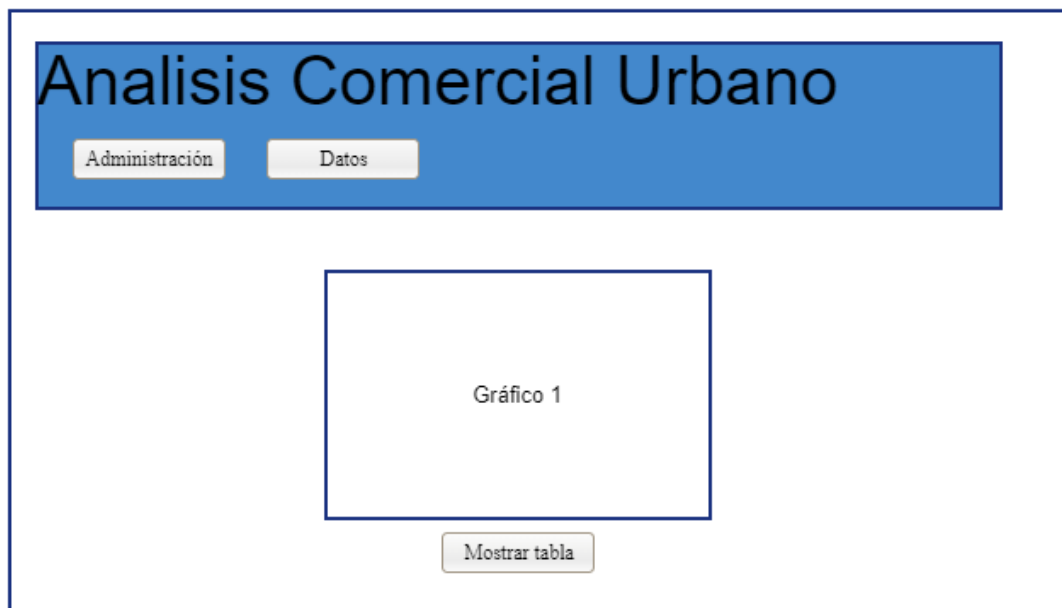


Figura C 7 Prototipo gráficos



Figura C 8 Prototipo tablas

Prototipo de interfaz de usuario para "Análisis Comercial Urbano". El encabezado azul contiene los botones "Administración" y "Datos". Debajo, hay una tabla con una fila de encabezado gris que dice "Tabla asociada a todos los datos" y tres filas de datos vacías.

<input type="checkbox"/>	Tabla asociada a todos los datos

Figura C 9 Prototipo tabla con todos los datos

Prototipo de interfaz de usuario para "Análisis Comercial Urbano". El encabezado azul contiene los botones "Administración" y "Datos". Debajo, hay dos botones: "Extraer datos" y "Subir datos".

Figura C 10 Prototipo menú administración

Apéndice D. Documentación Técnica de Programación

D.1. Introducción

En este apéndice describiré la documentación técnica relacionada con el desarrollo del proyecto.

D.2. Estructura de directorios

Esta es la estructura de ficheros alojados en:

https://github.com/SergioBueno27/TFG_GII_Analisis_Comercial_Urbano

La aplicación se distribuye de la siguiente forma:

- **src:** Contiene el proyecto Symfony principal
- **db:** Contiene la copia de la base de datos
- **vh:** Contiene el fichero de configuración virtualhost de Apache
- **img:** Contiene las imágenes usadas en la documentación
- **doc:** Contiene la documentación del proyecto tanto memoria como anexos
- **src/bin:** Contiene binarios de Symfony
- **src/config:** Contiene la configuración de Symfony
- **src/csv :** Contiene ficheros generados al extraer los datos en formato csv
- **src/csv copy:** Contiene ficheros copia de los ficheros csv
- **src/public:** Contiene los ficheros públicos de la aplicación
- **src/public/css:** Contiene los ficheros css de la aplicación
- **src/public/icon:** Icono de la aplicación
- **src/public/sweetalert2:** Librerías de sweet alert 2
- **src/sass:** Contiene los ficheros Sass de la aplicación
- **src/scripts:** Scripts de la aplicación
- **src/templates:** Contiene los ficheros de la parte de vista del modelo
- **src/translations:** Contiene los ficheros de traducción de la aplicación
- **src/src/Controller:** Contiene los controladores de la aplicación
- **src/src/Entity:** Contiene las entidades de nuestro proyecto vinculadas a nuestra base de datos
- **src/src/Form:** Contiene los formularios de la aplicación
- **src/src/Migrations:** Contiene las migraciones realizadas a través de doctrine
- **src/src/Repository:** Contiene los repositorios desde los que realizar peticiones a la base de datos

- **src/src/Migrations:** Contiene las migraciones realizadas a través de doctrine
- **src/src/Security:** Contiene la seguridad de autenticación de Symfony
- **src/src/Service:** Contiene los servicios de nuestra aplicación

D.3. Manual del programador

Instalación

En primer lugar necesitamos un sistema operativo pudiendo usar cualquier sistema operativo pero en el caso de este proyecto se decidió por Linux más concretamente la distribución Ubuntu. Los pasos de instalación son los siguientes:

Primero se instala PHP7.3:

```
sudo apt install php7.3 php7.3-cli php7.3-mbstring
```

Posteriormente se instalan los paquetes necesarios para instalar *composer*:

```
sudo apt install curl git unzip
```

Se recoge el instalador usando curl:

```
curl -sS https://getcomposer.org/installer -o composer-setup.php
```

Igualmente será necesaria una base de datos sobre la que almacenar los datos, se podrá elegir de igual forma MySQL o MariaDB. Para MySQL:

```
sudo apt update
sudo apt install mysql-server
sudo mysql_secure_installation
```

Se instala Node.js junto con el gestor de paquetes NPM para instalar las librerías empleadas de esta fuente:

```
curl -sL https://deb.nodesource.com/setup_13.x | sudo -E bash -
sudo apt-get install -y nodejs
```

Se verifica que el instalador se ha descargado sin estar corrupto:

```
php -r "if (hash_file('SHA384', 'composer-setup.php') ===
'669656bab3166a7aff8a7506b8cb2d1c292f042046c5a994c43155c0be6190
fa0355160742ab2e1c88d40d5be660b410') { echo 'Installer verified'; } else {
echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
```

Se instala *composer* de forma global


```
sudo php composer-setup.php --install-dir=/usr/local/bin --  
filename=composer
```

Se instala Symfony

```
wget https://get.symfony.com/cli/installer -O - | bash
```

Posteriormente se puede emplear la herramienta de Symfony que verifica que paquetes o módulos PHP extra son necesarios

```
symfony check:requirements
```

A través de esta lista se instalar el resto de los paquetes restantes.

Por último se clona el proyecto en local:

```
https://github.com/SergioBueno27/TFG_GII_Analisis_Comercial_Urbano
```

Ejecución en local

Para la ejecución en local serán necesarios los siguientes pasos:

Instalamos los paquetes de NPM y *composer* desde la carpeta donde se haya clonado el proyecto para ello se puede emplear el script de la carpeta `src/scripts/main.sh`:

```
npm install  
composer install
```

Posteriormente será necesario ejecutar el script de la carpeta `src/scripts/linkNodeModules.sh`:

```
ln -rs ../node_modules/@mdi ../public/@mdi  
ln -rs ../node_modules/ag-grid-community ../public/ag-grid-community  
ln -rs ../node_modules/chart.js/dist/ ../public/chart.js  
ln -rs ../node_modules/sweetalert2/dist/sweetalert2.all.min.js  
../public/sweetalert2/sweetalert2.all.min.js
```

Como servidor web se puede emplear el incorporado dentro de Symfony o como en el caso de este proyecto usar APACHE o cualquier otro servidor WEB.

También hay que destacar que una vez extraídos los datos provenientes de la API será necesario mover los ficheros descargados a la carpeta donde se encuentre nuestra base de datos, para posteriormente subirlos, para este caso se puede emplear el script de la carpeta `src/scripts/copycsv.sh`:

```
sudo cp ../csv/*.csv /var/lib/mysql/Proyecto/
```

Despliegue en Cloud VPS

Para el despliegue de la aplicación me he decidido por emplear un servidor en la nube, para ello me he decidido por este servicio <https://www.arubacloud.com/>. El coste mensual de la aplicación ha sido solo de 3.38 €, con recursos más que suficientes para nuestra aplicación.

Los pasos para la instalación han sido los siguientes:

Se actualiza el listado de paquetes:

```
yum update
```

Se instala Apache como servidor web:

```
yum install httpd  
sudo systemctl enable --now httpd
```

Se añade el repositorio desde el que descargar PHP:

```
sudo yum -y install http://rpms.remirepo.net/enterprise/remi-release-7.rpm  
sudo yum -y install epel-release yum-utils
```

Desactivar PHP 5.4 por defecto y seleccionar 7.3:

```
sudo yum-config-manager --disable remi-php54  
sudo yum-config-manager --enable remi-php73
```

Instalar módulos PHP necesarios para uso de la base de datos y Symfony:

```
sudo yum -y install php php-cli php-fpm php-mysqlnd php-zip php-devel  
php-gd php-mcrypt php-mbstring php-curl php-xml php-pear php-bcmath  
php-json php-mysql php-mysqlnd
```

Añadir repositorio MariaDB, en el despliegue nos hemos decantado por el uso de MariaDB puesto que la funcionalidad es igual a MySQL y su instalación da menos problemas en sistemas operativos CentOS:

```
sudo tee /etc/yum.repos.d/MariaDB.repo<<EOF  
[mariadb]  
name = MariaDB  
baseurl = http://yum.mariadb.org/10.4/centos7-amd64  
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB  
gpgcheck=1  
EOF
```

Se actualiza el listado de paquetes

```
sudo yum makecache fast
```

Se instala MariaDB y se activa el servicio de la base de datos:

```
sudo yum -y install MariaDB-server MariaDB-client
```

```
sudo systemctl enable --now mariadb
```

```
sudo mysql_secure_installation
```

Se sube el proyecto Symfony al servidor en el caso del proyecto /var/www/public/ y se configuran las variables del fichero .env con respecto a los nuevos parámetros del servidor.

Se recupera el fichero de configuración de Apache (incluido en el repositorio del proyecto vh/AnalisisComercial.conf) y se incluye en la carpeta del servidor /etc/httpd/conf.d/

En este punto se puede importar la base de datos del fichero (db/db_copy.sql)

Por último habilitamos y configuramos el módulo de seguridad SELinux para ello primero cambiamos el usuario y grupo de las carpetas del proyecto:

```
chown apache:apache -R /var/www/public
```

Se habilita el acceso del sistema para leer y escribir en la carpeta del proyecto:

```
chcon -R -t httpd_sys_rw_content_t /var/www/public
```

Se permite el acceso a la base de datos:

```
setsebool -P httpd_can_network_connect_db=1
```

Y se habilita el modulo SELinux

```
setenforce 1
```

Apéndice E. Documentación de Usuario

E.1. Introducción

En este manual se detallan aspectos de uso para usuarios de la aplicación.

E.2. Requisitos de usuarios

Se garantiza el correcto uso en navegadores:

- Mozilla Firefox versión 51
- Google Chrome versión 56
- Internet Explorer versión 11
- Microsoft Edge versión 38

Se puede acceder desde otros navegadores pero la comprobación se ha realizado sobre estos.

Igualmente se ha comprobado el uso desde dispositivos móviles, en concreto Firefox Browser versión 68 y chrome.

E.3. Instalación

Al ser una aplicación web no requiere de ninguna instalación, solo en caso de que se decida por instalar en local, teniendo que emplear el manual definido previamente.

A la aplicación se puede acceder desde el link:

<http://80.211.68.16/>

E.4. Manual de Usuario

Acceso a la aplicación

Como ya se ha descrito para acceder a la aplicación se puede usar el link:

<http://80.211.68.16/>

Página de acceso

Nada más acceder a la página se redirige junto con el idioma del navegador a la página de acceso, en caso de existir dentro de los idiomas disponibles se asigna por defecto y en caso contrario se elige inglés por defecto.



Figura E 1 Página de acceso

Página de registro

Antes de permitir el acceso a la aplicación se requiere registrar un usuario a la aplicación.



Figura E 2 Página de registro

Cambio de idioma

En caso de querer cambiar el idioma actual:



Figura E 3 Cambio de idioma

Tanto la página de acceso, como en la página de registro , como en el menú principal, se permite cambiar el idioma de entre los disponibles.

Menú principal

Una vez se ha accedido a la aplicación se visualizan dos variantes del menú principal:

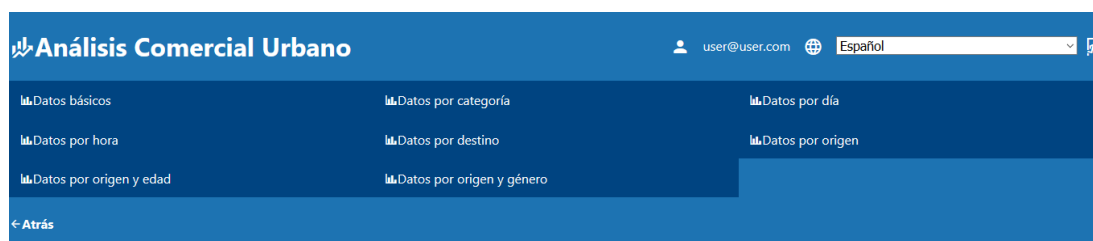


Figura E 4 Menú con rol de administrador

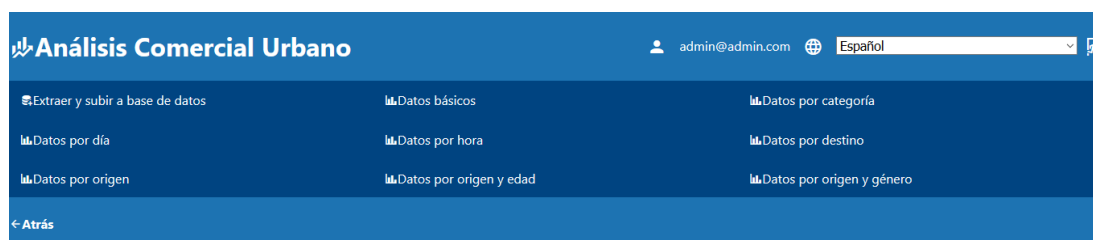


Figura E 5 Menú con rol de usuario

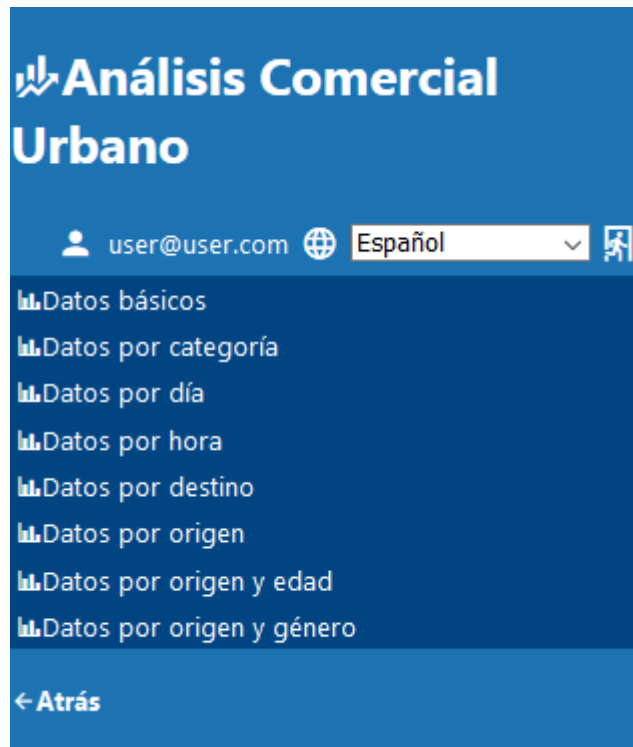


Figura E 6 Menú con rol de usuario

El primero se corresponde al usuario administrador de la aplicación que dispone de la opción de extraer y subir datos a la base de datos de la aplicación a través de peticiones a la API BBVA.

La segunda imagen nos permite visualizar el aspecto que un usuario normal obtiene al acceder a la aplicación.

Por último la última imagen, muestra el reescalado al emplear, una resolución menor a 1000px, permitiendo la visualización con dispositivos móviles.

Página de inicio

La página de inicio simplemente muestra el menú principal permitiendo seleccionar alguna de las opciones disponibles:

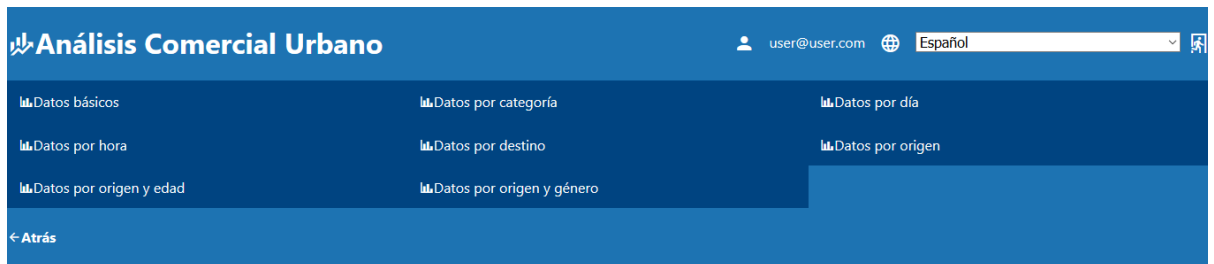
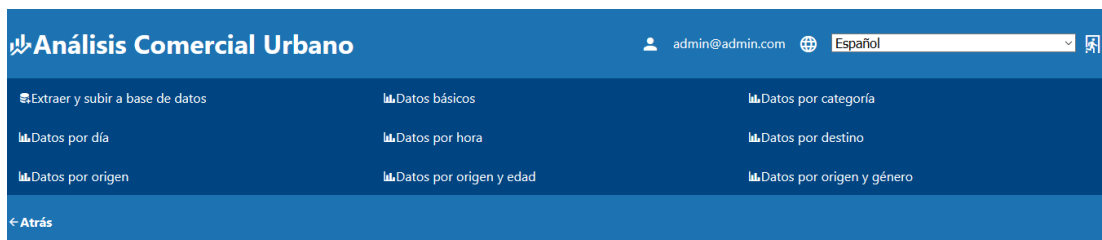


Figura E 7 Página de inicio

Al hacer *click* sobre cada una de las opciones se muestran los gráficos asignados al tipo de dato seleccionado.

Página de extraer y subir datos

Al acceder como administrador se permite extraer datos de la API BBVA y subir los datos sobre nuestra base de datos:



Recuerde que algunas operaciones pueden durar varios minutos

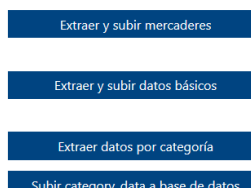


Figura E 8 Página de administración

Al acceder como administrador nos permite extraer y subir por cada tipo de dato.

Gráficos

Como ya se ha mencionado al hacer *click* sobre cada botón del menú se visualizan los gráficos asociados al tipo de dato seleccionado:

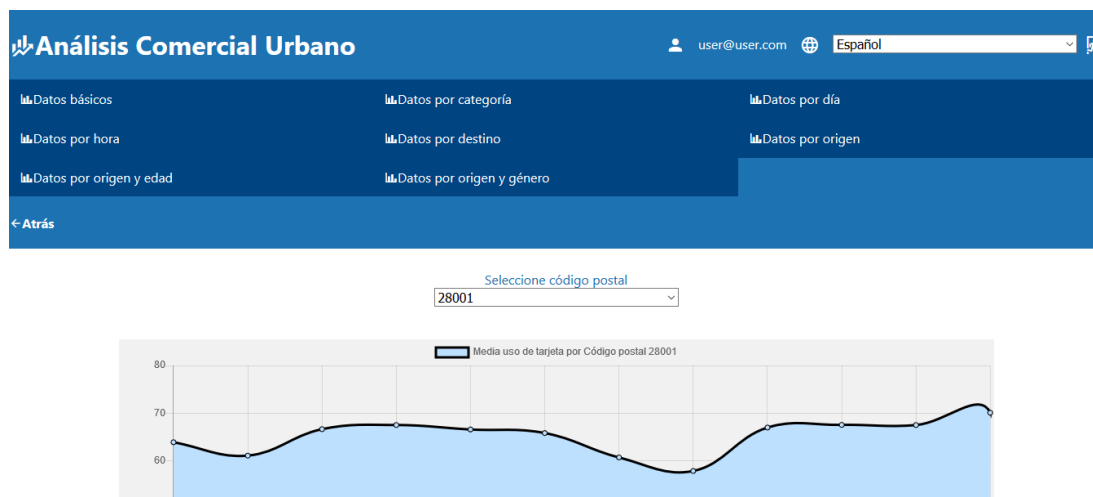


Figura E 9 Datos básicos



Figura E 10 Datos por categoría

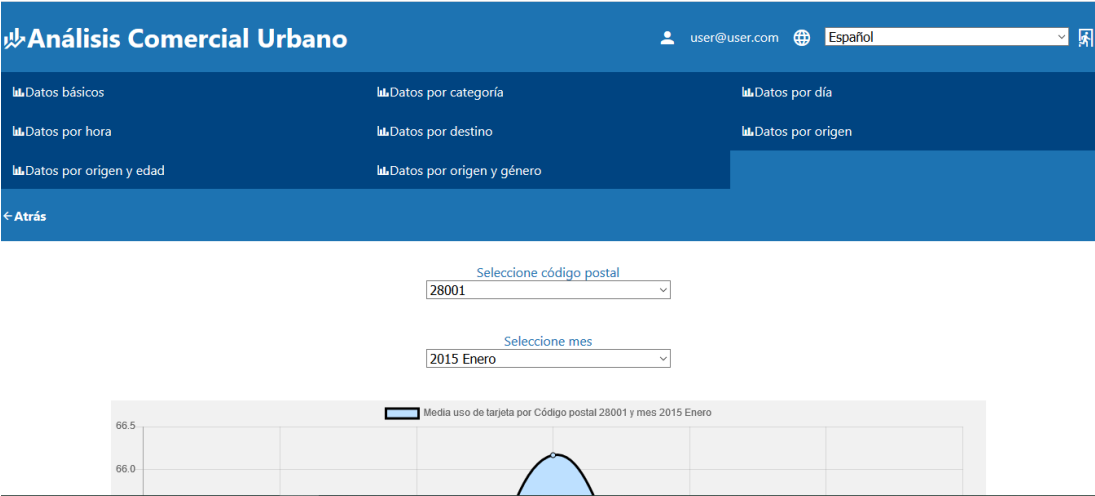


Figura E 11 Datos por día

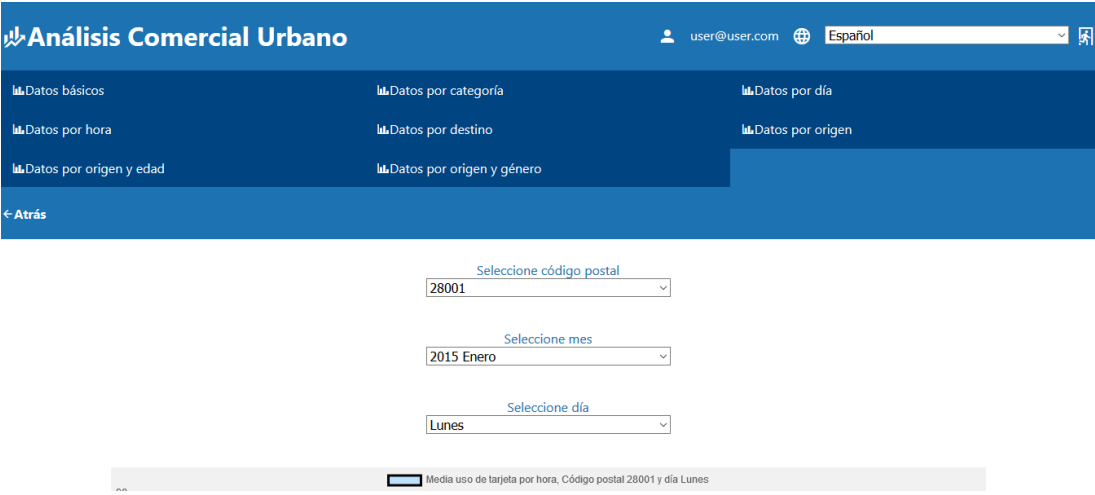


Figura E 12 Datos por hora

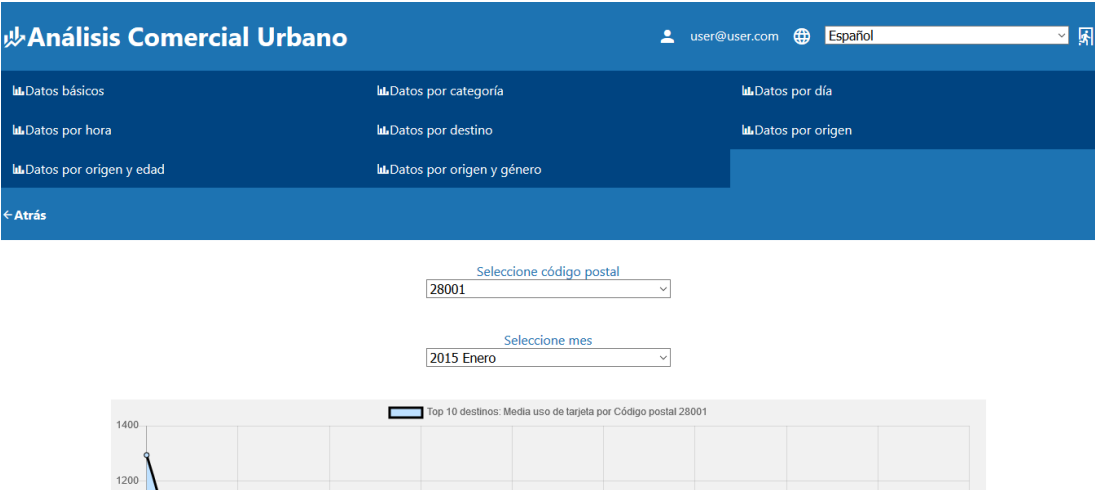


Figura E 13 Datos por destino

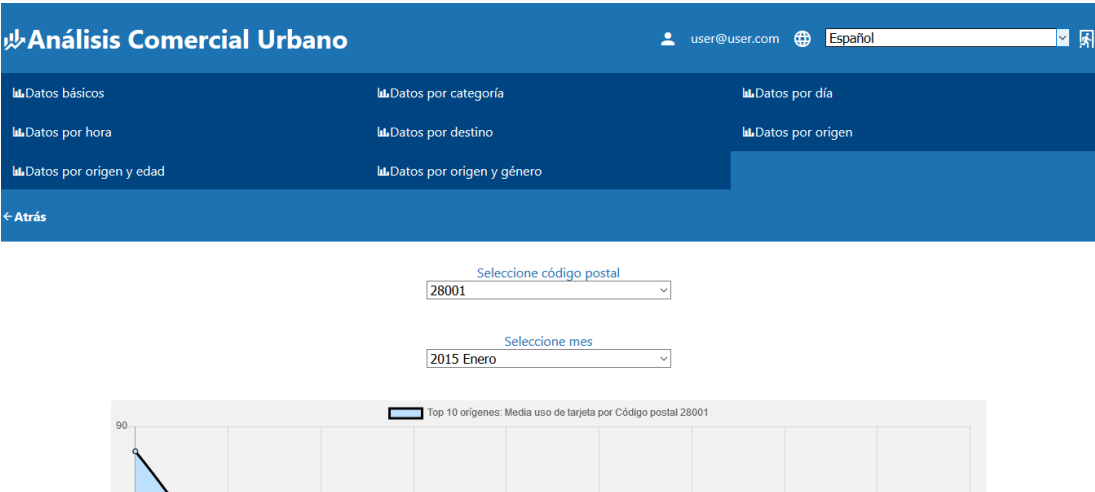


Figura E 14 Datos por origen

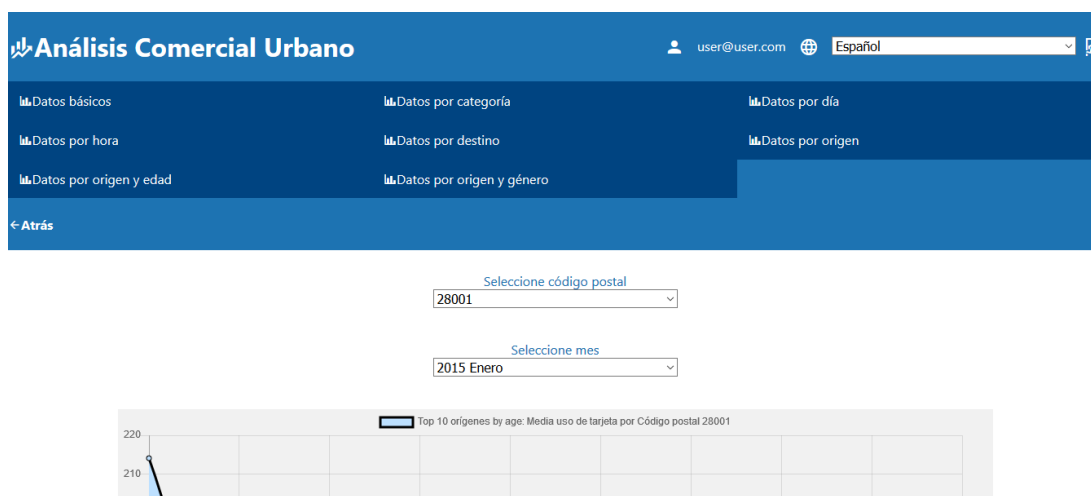


Figura E 15 Datos por origen y edad

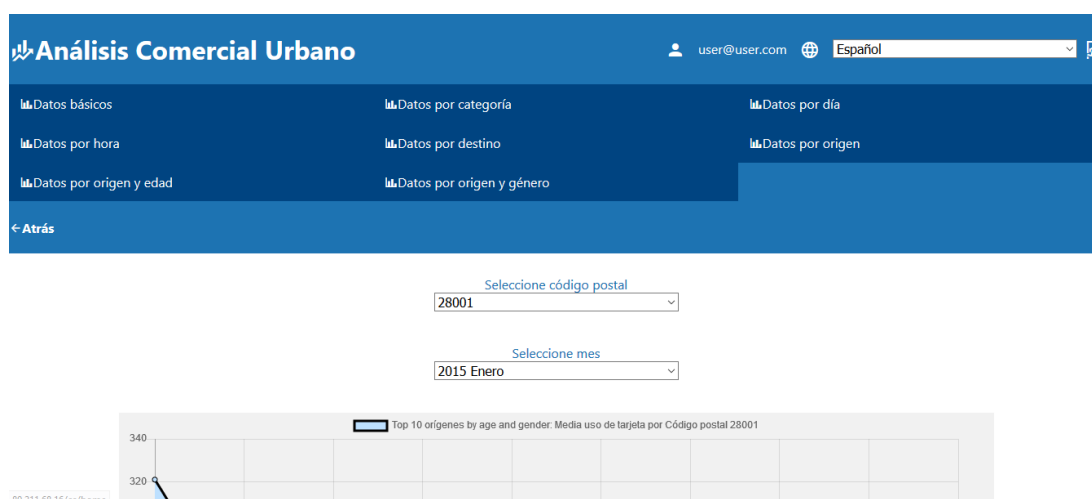


Figura E 16 Datos por origen y género

Mostrar tabla asociada a los datos

Figura E 17 Mostrar tablas

Mostrar todos los datos

Figura E 18 Mostrar todos los datos

Descargar fichero .net

Figura E 19 Descargar fichero .net

En todos se permite, seleccionar el código postal, sobre el que mostrar los datos y según el tipo de dato, seleccionar opciones adicionales como son el código de categoría, el mes y el día.

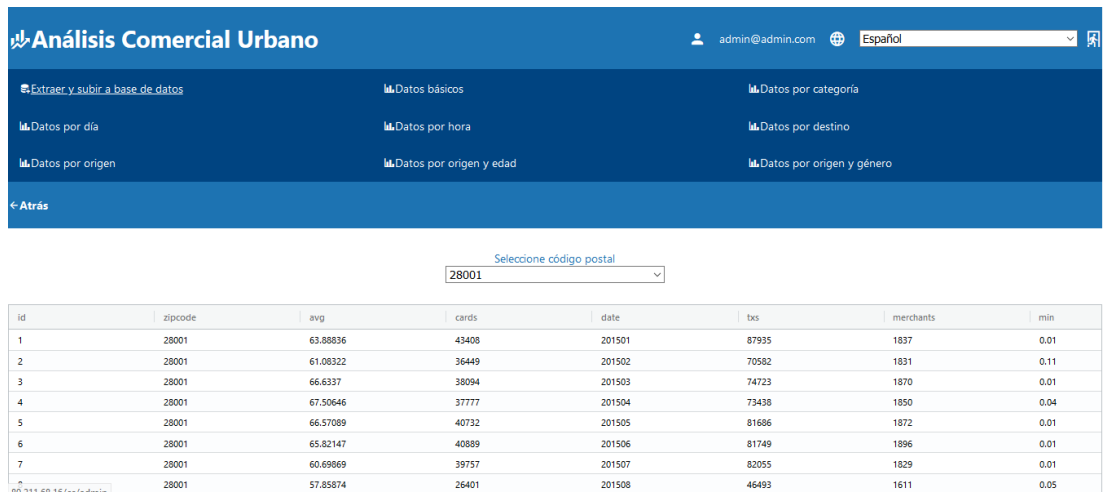
Después de mostrar los gráficos, se pueden ver las tablas asociadas, al código postal, empleando el botón de la figura anterior.

Para que aparezca el último botón “Mostrar todos los datos” se debe acceder en modo administrador, en este caso se mostrará una tabla con todos los códigos postales en lugar de mostrar de uno solo.

Por último se ha implementado el botón “Descargar fichero .net” que permite descargar el contenido de las tablas de origen y de destino en un formato compatible con visores de grafos para obtener información extra.

Tablas

Al pulsar el botón “Mostrar tabla asociada a los datos” se muestra una tabla con los datos del código postal actual, adicionalmente permite cambiar de código postal de la misma forma que desde los gráficos:



id	zipcode	avg	cards	date	txs	merchants	min
1	28001	63.88836	43408	201501	87935	1837	0.01
2	28001	61.08322	36449	201502	70582	1831	0.11
3	28001	66.6337	38094	201503	74723	1870	0.01
4	28001	67.50646	37777	201504	73438	1850	0.04
5	28001	66.57089	40732	201505	81686	1872	0.01
6	28001	65.82147	40889	201506	81749	1896	0.01
7	28001	60.69969	39757	201507	82055	1829	0.01
8	28001	57.85874	26401	201508	46493	1611	0.05

Figura E 20 Tablas

Descargar como csv

Igualmente disponemos de un botón que nos permite descargar la tabla en formato csv, por si fuera necesario para un análisis posterior:

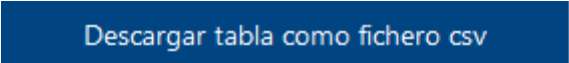
Un botón rectangular de color azul oscuro con el texto "Descargar tabla como fichero csv" en color blanco.

Figura E 21 Descargar como csv

Este botón aparece después de todas las tablas de la aplicación

Desplazamiento

Se dispone de diversos botones para facilitar la navegabilidad dentro de la aplicación:



Figura E 22 Salir de la aplicación

Un botón rectangular de color azul oscuro con el texto "← Atrás" en color blanco.

Figura E 23 Atrás