TRABAJOS FIN DE GRADO

Universidad de Jaén

*Escuela Politécnica Superior de Jaén*

# Paquete R para el análisis de datos bibliométricos en publicaciones científicas

Autor: Juan Miguel Jiménez Rivas

Grado: Ingeniería en Informática

Director: Francisco Charte Ojeda
Departamento del director: Informática

Fecha: 08/09/2024

UNIVERSIDAD DE JAÉN

D./Dª Francisco Charte Ojeda, tutor(es) del Trabajo Fin de Grado titulado: **Paquete R para el análisis de datos bibliométricos en publicaciones científicas**, que presenta Juan Miguel Jiménez Rivas, autoriza(n) su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, 08/09/2024

El estudiante

JIMENEZ RIVAS JUAN MIGUEL - 50642691A

Firmado digitalmente por JIMENEZ RIVAS JUAN MIGUEL - 50642691A
Fecha: 2024.09.08 20:44:14 +02'00'

Juan Miguel Jiménez Rivas

El tutor

Fecha: 2024.09.09 08:14:18 +02'00'

Francisco Charte Ojeda

**Agradecimientos**

En primer lugar, deseo expresar mi más sincero agradecimiento a mis familiares y amigos por su incondicional apoyo en estos momentos, ayudándome a continuar y enfocarme en este proyecto, el cual ha sido realizado con gran esfuerzo y, en ocasiones, alguna que otra lágrima. Sin vosotros, esto no habría sido posible. Os quiero profundamente.

A mis padres y a mi hermano, gracias por confiar en mí incluso cuando yo mismo no lo hacía. Siempre habéis visto en mí capacidades que ni siquiera yo creía tener, y tal vez por eso hoy estoy entregando este trabajo. Sé cuánto me queréis y cuánto habéis hecho para que llegue hasta aquí. De corazón, gracias papá y mamá, espero poder devolveros algún día todo lo que habéis hecho por mí.

También quiero agradecer a una persona muy especial que, a lo largo de este proyecto, ha sido un apoyo fundamental. Sus opiniones sobre el diseño y sus consejos han sido invaluables, especialmente en los momentos en los que estuve al borde de rendirme. Eres una persona muy importante para mí y siempre lo serás. Te quiero mucho María.

Por último, pero no menos importante, quiero agradecer a mi tutor, el profesor Francisco Charte, por su apoyo constante. Siempre ha estado disponible cuando lo he necesitado, revisando mi trabajo y dándome recomendaciones sobre cómo mejorarlo. Desde un principio, tuve claro que quería que el tutor de mi trabajo de fin de grado fuera un gran profesor que me hubiese marcado a lo largo de la carrera, y no tuve ninguna duda en elegirle a usted. Gracias por el amor y dedicación que pone en su trabajo, un ejemplo que inspira a los alumnos a estudiar y prepararse con otro enfoque.

Gracias a todos, de corazón.

# Tabla de contenidos

# Lista de figuras

# Lista de tablas

# Lista de listados de código

# Capítulo 1

# INTRODUCCIÓN Y MOTIVACIÓN

Este capítulo introductorio dará una visión general sobre el objetivo, la razón y la manera de llevar a cabo el desarrollo del TFG.

El proyecto implica desarrollar e implementar un paquete en R para analizar datos bibliométricos de publicaciones científicas. Para llevar a cabo este proyecto, he necesitado una revisión y puesta a punto sobre el lenguaje R y la manera de realizar paquetes para este lenguaje.

## 1.1. Introducción

Hoy en día el mundo de la investigación pasa por la publicación de artículos en revistas científicas. Según la organización WordsRated [1] en 2022 se publicaron más de cinco millones de artículos científicos, con un aumento constante de publicaciones desde 2018 del 22,78 %, lo que demuestra un incremento exponencial en el número de investigaciones con el paso de los años.

Por tanto, los datos que se pueden obtener de la publicación de una actividad científica ganan gran importancia, ya que estos permiten obtener indicadores tanto de la misma, como de cada uno de los autores de ella, con estos datos se puede llevar a cabo una evaluación de la divulgación.

Estos datos usados como una herramienta, se le llama **bibliometría**. Esta herramienta, ayuda a los investigadores a la toma de decisiones. Hoy en día, con el número elevado de publicaciones, el entorno se ha vuelto más competitivo. Por tanto, se nece-

sita mejor rendimiento de los investigadores. Para ello la principal baza es el manejo de las mejores fuentes de información.

Con los avances actuales, las plataformas y bases de datos más usadas y reputadas como: Web Of Science (WOS), Scopus, entre otras, ofrecen estos datos bibliométricos. Estos datos no solo sirven para poder comparar y evaluar como se ha mencionado anteriormente, también permite conocer cómo y dónde se citan y utilizan estas publicaciones científicas.

## 1.2. Motivación

El principal problema que enfrentan hoy en día los investigadores al buscar información no es solo el tiempo que deben invertir en esta tarea, sino también la cantidad de publicaciones existentes y generadas constantemente. Esto les obliga a realizar un cribado exhaustivo para identificar cuáles son realmente relevantes.

Para obtener publicaciones destacadas, es recomendable considerar varios factores:

- Reputación del lugar de publicación, es preferible optar por revistas con mayor prestigio, ya que suelen mantener estándares de calidad que avalan su estatus
- Posición del artículo en la revista, es más adecuado elegir publicaciones bien posicionadas en los rankings, en lugar de aquellas que ocupan posiciones bajas o que no están clasificadas
- Número de citas, aunque número absoluto puede ser engañoso debido a la antigüedad de la publicación.
- Promedios de citas, para evitar errores debido a lo explicado en el ítem anterior, se utilizan medidas como el FWCI (Índice de Citación Ponderado por Campo), el índice h o índice 10 para comparar entre publicaciones, entre otros.

Buscando el ahorro de tiempo para estos investigadores y aprovechando la facilidad con la que hoy en día se pueden obtener colecciones de datos públicos gracias a un conjunto de métodos que permiten obtener la información de estos bancos de datos, el principal objetivo del presente trabajo, tras consultar en CRAN[1] y comprobar

[1] Repositorio de paquetes de R https://cran.r-project.org

que no existe ningún paquete que realice una demostración y análisis de datos bibliométricos, se llevará a cabo el diseño e implementación de un paquete para R. Este pretende facilitar la visualización e interpretación de información desde las bases de datos mencionadas, pudiendo añadirse fácilmente más bases de datos. Primeramente realizando consultas sobre cada uno de estos conjuntos de datos y recogiendo la información y llevando a cabo un análisis de los mismos.

## 1.3. Objetivos

El objetivo general de este trabajo es llevar a cabo un paquete que cumpla las funciones explicadas justo en el apartado anterior, aunque los objetivos que se perseguirán más específicamente son:

- Investigar y comprender el concepto de la bibliometría
- Revisar paquetes existentes en R
- Aprender fundamentos de R, mediante documentación y libros
- Instalar software necesario para utilización y desarrollo del paquete R
- Implementar funciones de análisis en R
- Crear interfaz gráfica para el usuario
- Documentar el paquete
- Verificar el paquete

## 1.4. Estructura de la memoria

La memoria se estructura en los siguientes apartados:

- Capítulo 1: el objetivo de este capítulo es captar el interés del lector sobre el problema al que se busca dar solución. Se presenta una introducción y se explican los objetivos que se pretenden alcanzar con este proyecto

- Capítulo 2: se ofrece una explicación preliminar antes de la selección de la metodología de trabajo, concluyendo con una sección dedicada a la estimación temporal y económica del proyecto
- Capítulo 3: en este capítulo se detalla el análisis del proyecto, describiendo todos los requisitos que debe cumplir. Además, se comentan las herramientas utilizadas a lo largo del proceso de desarrollo
- Capítulo 4: aquí se planificará la gestión temporal del proyecto, estableciendo un cronograma detallado.
- Capítulo 5: Este capítulo se centrará en el diseño previo a la implementación, proporcionando una guía detallada que permitirá al programador seguir un esquema definido y evitar depender de decisiones subjetivas. De este modo, se garantiza que el diseño sea seguido de manera rigurosa y coherente durante la implementación
- Capítulo 6: se abordará el diseño visual de la interfaz gráfica, mostrando el proceso desde los primeros bocetos hasta la definición del logotipo, la paleta de colores y las tipografías
- Capítulo 7: se ofrecerá una explicación detallada del desarrollo del paquete, incluyendo las tecnologías y las fuentes de datos utilizadas. También se describirá la implementación, proporcionando una visión completa del proceso y las decisiones tomadas
- Capítulo 8: este capítulo será un manual de uso que permitirá a cualquier persona con un ordenador utilizar el paquete de manera sencilla

# Capítulo 2

# METODOLOGÍA Y ESTIMACIÓN

Tras establecer los objetivos del proyecto, otro aspecto importante es seleccionar una metodología para su desarrollo y elaborar una planificación previa. En este capítulo se expondrá la metodología a seguir y se proporcionará la estimación de planificación previa elaborada.

## 2.1. Metodología

Existen varias metodologías de trabajo y planificación de proyectos vistas en el grado, cada una diseñada para adaptarse a diferentes tipologías de proyectos. Esto implica la necesidad de revisar cada metodología y evaluar cuál se adaptará mejor para la realización del trabajo fin de grado.

### 2.1.1. Modelos tradicionales

Los modelos tradicionales se caracterizan por seguir un enfoque secuencial y planificado para el desarrollo de proyectos. Estos modelos se distinguen por la fijación estricta de requisitos al comienzo del proyecto. Por lo general son metodologías poco flexibles.

#### Modelo en cascada [2]

Es un modelo más antiguo, el cual es secuencial y divide el proceso en etapas:

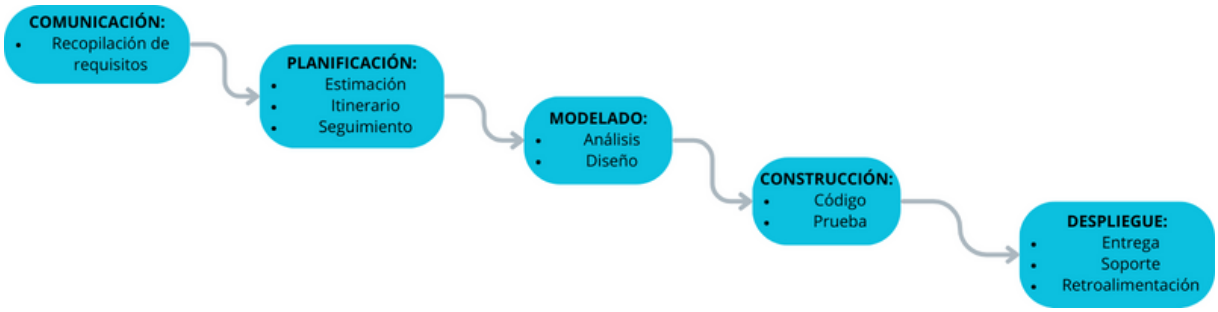



Figura 2.1: Diagrama de las etapas de modelo en cascada.
**Fuente: Elaboración propia.**

**Ventajas:**

1. Modelo lineal y sencillo

2. Si el modelo se adapta correctamente, es muy seguro respecto a errores

3. Permite mejores estimaciones tanto de calendarios, como de presupuestos

4. Puede adaptarse bien a mejoras bien definidas a un sistema ya existente

5. Se adapta bien a proyectos de pequeña envergadura o complejidad, donde los requisitos no cambian, siempre que estos estén bien definidos y sean estables

**Inconvenientes:**

1. Es un modelo lineal, por lo tanto, es inflexible a la hora de incorporar nuevos requisitos

2. Si los requisitos u objetivos no son claros, el proceso por regla general tendrá errores, los cuales serán difíciles de corregir

Se entiende que para el cliente puede ser complicado enunciar de manera explícita todos los requisitos. Sin embargo, en este modelo es necesario que lo haga, ya que cualquier cambio posterior podría generar confusión. Además, es importante que el cliente tenga paciencia, ya que no dispondrá de una versión funcional del programa hasta que el proyecto esté en una fase muy avanzada.

**Modelo incremental [2]**

El modelo incremental combina elementos de los flujos de proceso lineal[1] y paralelo[2]. Este modelo, en lugar de entregar el producto completo de una sola vez, realiza entregas en **incrementos**, lo que significa que cada entrega es un producto operativo, aunque incompleto.

En este modelo, normalmente, el primer incremento consistirá en el producto que cumple los requerimientos básicos, sin incluir muchas características adicionales.

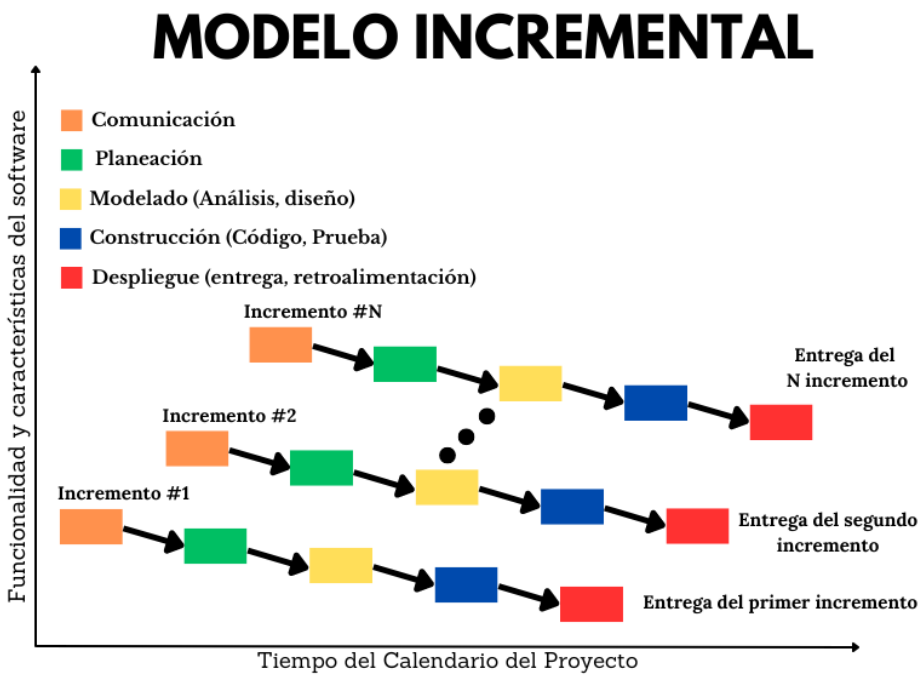



Figura 2.2: Diagrama de las etapas de modelo incremental.
**Fuente: Elaboración propia.**

Respecto a cada incremento, una vez ha comenzado el desarrollo de uno de ellos, sus requisitos no se modifican hasta que ha terminado. Los cambios se incluirán en la planificación de un incremento posterior.

**Ventajas:**

1. Los clientes pueden utilizar el software con características esenciales en las etapas principales

[1]Proceso lineal: sigue una secuencia paso a paso, cada fase debe completarse completamente antes de avanzar.
[2]Proceso paralelo: las tareas se pueden ejecutar simultáneamente, o al menos de forma parcialmente superpuesta

2. Los primeros incrementos pueden servir como prototipos, proporcionando información valiosa y siendo evaluados por el cliente
3. El riesgo de que el proyecto falle es mucho menor en comparación con el modelo en cascada
4. Los incrementos principales son los más probados

**Inconvenientes:**

1. Dificultad de determinar en qué incremento alojar algunos requisitos
2. Dificultad para determinar parte común

Este modelo es mucho más acorde a este proyecto. Aunque también tiene inconvenientes, que no se adaptan a al proyecto y otros modelos se adaptan mejor.

### Modelo DRA (Desarrollo Rápido de Aplicaciones) [3]

Este modelo, similar al enfoque del modelo en cascada, se caracteriza por su alta velocidad. Se centra en la rápida entrega de prototipos, utilizando componentes existentes y creando otros reutilizables. Los sistemas basados en este modelo han tenido un gran éxito, ya que muchos proyectos comparten numerosas partes en común.

**Ventajas:**

1. Permite un desarrollo rápido de aplicaciones sencillas
2. Partes de código ya implementado

**Inconvenientes:**

1. En sistemas grandes, puede ser complejo organizar a los equipos
2. Difícil implementar en proyectos no estándares, con código ya implementado
3. Problemas de memoria al cargar código que implemente ciertas funcionalidades no necesarias

El modelo DRA no se considera adecuado a este proyecto debido a que no existe implementación de código disponible.

### 2.1.2. Metodologías ágiles

Este tipo de metodologías son las más usadas por la mayoría de empresas privadas como Santander [4], entre otras muchas. Esto es así, ya que este tipo de metodología, se centra en la entrega continua de producto al cliente y adaptación rápidamente de cambios que se piden. Estas metodologías fomentan colaboración y retroalimentación contínua.

#### Scrum [5]

Es el método ágil más aplicado, que se basa en *Sprints*[3] . Tras cada iteración se entrega un producto. Posteriormente, se muestra al cliente y este opina sobre el producto. Esta metodología está desarrollada para la realización del proyecto en equipo. Por tanto, tras esta reunión con el cliente, debería de producirse otra reunión del equipo a solas. El objetivo de la reunión es la de ir mejorando tras cada iteración. Esta metodología también da bastante peso al ritmo de trabajo tanto día a día, como en cada iteración. Llevando a cabo en todo momento una estimación adecuada. Lo importante es una fecha clave de iteración muy estable. La fecha fin de iteración será cuando se producirá la reunión con el cliente-tutor. El ciclo de trabajo a seguir por esta metodología es el siguiente:

- **Product Backlog**: requisitos del cliente priorizados y estimados
- **Sprint Backlog**: selección de requisitos del Product Backlog, ya descompuestos en tareas
- **Burndown Chart**: gráficas que representan el trabajo pendiente tanto del sprint, como del proyecto completo

También esta metodología presenta una serie de reuniones :

- **Daily Meeting**: reunión breve de 10-15 minutos con el equipo para evaluar lo realizado el día anterior, planificar las tareas del día y discutir posibles obstáculos
- **Planificación del sprint**: en esta reunión se planifica el próximo *sprint*, definiendo claramente el objetivo de la iteración y desglosándolo en tareas específicas

[3] iteraciones cortas de tiempo (entre 1 y 4 semanas)

- **Revisión del Sprint**: se realiza un análisis informal del trabajo realizado con el tutor o cliente, buscando su opinión y retroalimentación continua sin que la reunión consuma mucho tiempo
- **Retrospectiva del equipo (Sprint Retrospective)**: tras la revisión del *sprint*, el equipo se reúne para identificar áreas de mejora y analizar cómo aumentar la productividad en futuros *sprints*

Este proyecto será llevado a cabo de manera individual y no en equipo. Por tanto, se pierde la ventaja del trabajo en equipo, que se adquiere con esta metodología. Sin embargo, aún se puede obtener beneficios al entregar producto en cada iteración, teniendo en cuenta en todo momento la opinión del tutor-cliente.

### Kanban [5]

Kanban es un método ágil que se basa en un tablón visual. Utiliza tarjetas o etiquetas visuales para facilitar la visualización del progreso, y se aplica tanto en los sprints como en el proyecto general. Kanban es particularmente útil para gestionar productos con requisitos en constante cambio o nuevas necesidades, y está diseñado para adaptarse a las variaciones en las prioridades de los requisitos. Es crucial que el tablón se mantenga actualizado en todo momento.

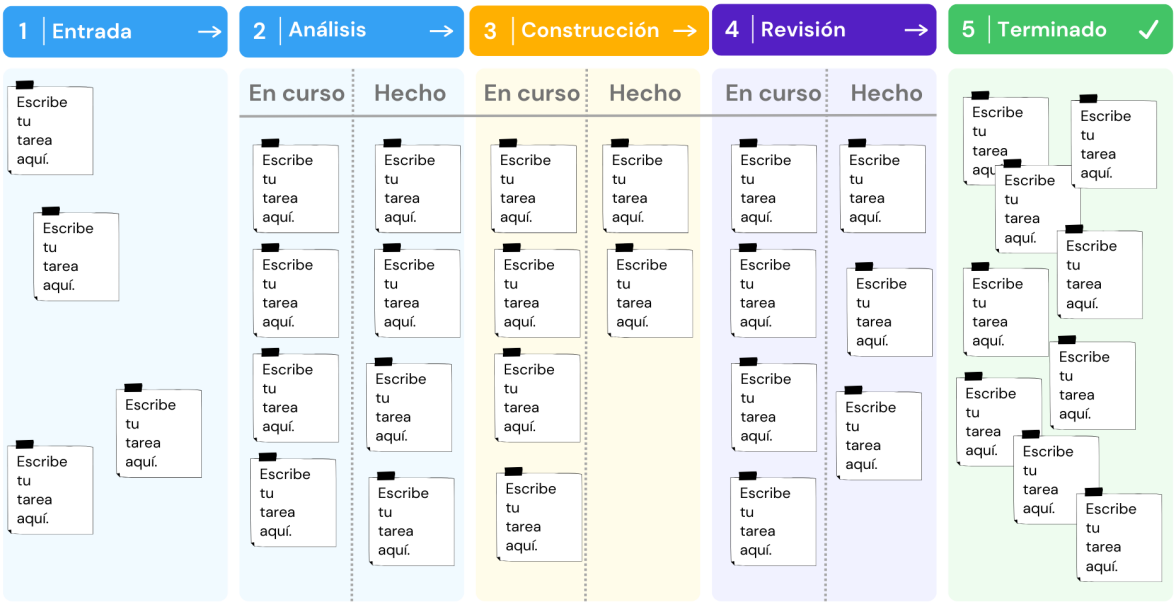



Figura 2.3: Representación de tablero KANBAN
**Fuente: Elaboración propia.**

La primera columna del tablón representa el *backlog* del método Scrum del producto, es decir, los requisitos y actividades pendientes por realizar. Se busca mantener una carga de trabajo similar en cada tarjeta.

### 2.1.3. Elección de método de trabajo

Después de un análisis exhaustivo de metodologías, tanto clásicas como ágiles, se ha decidido llevar a cabo el proyecto mediante un enfoque híbrido que combina elementos de Scrum y Kanban. Este enfoque híbrido se caracteriza por tener un *sprint* fijo en el tiempo, durante la finalización de los cuales se obtendrá retroalimentación del tutor-cliente en cada reunión. La integración de Kanban permitirá visualizar de manera clara los estados de cada tarea o elemento del proyecto, esto facilitará la identificación de obstáculos. Además, este enfoque híbrido permite priorizar tareas más importantes y reduce el riesgo de incumplimiento de plazos.

## 2.2. Estimación

La realización del proyecto comienza el 21 de marzo y finalizará, el 2 de septiembre, teniendo 162 días (5,39177 meses) para su desarrollo, realizando un trabajo de media jornada al día. Previamente, se realizaron tareas de documentación y lectura tanto de documentación del lenguaje R, como de APIs relacionadas con el proyecto.

### 2.2.1. Estimación de recursos

La responsabilidad total de este proyecto será del estudiante Juan Miguel. Por tanto se encargará tanto del análisis del proyecto, como de la implementación. Para estimar los costes de recursos necesitaremos:

- Equipo informático:
- Visual Paradigm:
    - Versión: 17.1 standard
- Alquiler:
    - Luz

| Componente | Especificación |
|---|---|
| Modelo | ASUS Zenbook 14 UX425EA-KI915 |
| Procesador | Intel Core i7-1165G7 |
| Memoria RAM | 16 GB DDR4 |
| Almacenamiento | 512 GB SSD |
| Sistema operativo | Windows 10 |
| Conectividad | Wi-Fi 6, Bluetooth 5.0 |
| Puertos | USB 3.2, HDMI |

Tabla. 2.1: Especificaciones de equipo informático

  - Agua
  - Internet: fibra óptica 500 MB

- Salario
- Material de oficina

### 2.2.2. Estimación de costes

#### Costes de hardware y software amortizable

| Concepto | Importe unitario | Amortización | Total/mes |
|---|---|---|---|
| ASUS Zenbook 14 UX425EA-KI915 | 1 088,85€ | 5 años | 18,15€ |
| Windows 11 Home | 109,90€ | 5 años | 1,83€ |
| **TOTAL** | | | **109,89€** |

Tabla. 2.2: Estimación de gastos amortizables

#### Costes de personal

Después de verificar la categoría salarial y el área a la que pertenece el trabajador, consultamos la siguiente resolución del Boletín Oficial del Estado [6].

Posteriormente, revisamos la última resolución [7] del convenio colectivo estatal de empresas de consultoría y tecnologías de la información. En esta resolución, se detallan las tablas salariales, donde muestran el salario bruto anual de un ingeniero en función de desarrollador, el cual asciende a **24 975,35€**. Realizando cálculos, el salario base mensual es de **1 655,11 €**. Por tanto, el precio de la hora sería: **10,35€/h**. La jornada será de 4 horas al día, por tanto 20 horas semanales, el importe mensual será de: **828€/mes**

| Concepto | Importe unitario | Duración | Total |
|---|---|---|---|
| Salario base, impuestos, seguro | 828,00€ | 5 meses y 11 días | 4 595,40€ |
| **TOTAL** | | | **4 595,40€** |

Tabla. 2.3: Estimación de gastos de personal

#### Costes adicionales

En este apartado se mostrarán los gastos tanto en recursos software no amortizables, como recibos de luz e internet.

| Concepto | Importe unitario | Duración | Total |
|---|---|---|---|
| Recibos de luz | 52,00€ | 6 meses | 312,00€ |
| INTERNET Fibra SMART 500Mb DIGI | 15,00€ | 6 meses | 90,00€ |
| Licencia Visual Paradigm Professional | 32,80€ | 6 meses | 196,80€ |
| | | **TOTAL** | **598,80€** |

Tabla. 2.4: Estimación de gastos adicionales

#### Estimación final

| Concepto | Importe |
|---|---|
| Costes de hardware y software amortizable | 109,89€ |
| Costes de personal | 4 595,40€ |
| Costes adicionales | 598,80€ |
| **TOTAL** | **5 304,09€** |

Tabla. 2.5: Estimación final de coste del proyecto

Los datos presentados representan una estimación exclusiva del coste del proyecto y no constituyen un presupuesto oficial. Además de estos costes estimados, deberían agregarse el IVA y otros gastos adicionales.

## 2.3. Estimación temporal

Es importante antes de nada realizar una estimación de fechas y tiempos de los recursos a utilizar.

El proyecto tuvo como comienzo al principio de marzo (**7 de marzo**), que tras reunión con el tutor-cliente que se lleva a cabo, en la cual se deducen conceptos básicos sobre el proyecto.

Tras dos meses desde la primera reunión, de puesta a punto del proyecto, durante los cuales se ha llevado a cabo el proceso inicial, que incluye la documentación, la selección de metodología de trabajo que se llevará a cabo, el conocimiento de lenguajes y herramientas para realizar este proyecto y un análisis de los requisitos establecidos por el cliente-tutor. El comienzo de desarrollo del software del proyecto comienza el día **7 de mayo**. También se fija como fecha de entrega, el día **9 de septiembre**, constando de 18 semanas. Dedicando al día unas 4 horas para así cumplir los objetivos plenamente. Por tanto se estiman que se dedicarán unas **360 horas** de desarrollo. Sin embargo, en caso de encontrar algún imprevisto, estas horas se podrán incrementar y decrementar. La planificación temporal detallada y el correspondiente diagrama de Gantt se desarrollarán en un capítulo aparte, dada su importancia. Este contenido se abordará en el capítulo 4.

# Capítulo 3

# ANÁLISIS

En este capítulo, se explicará el análisis detallado sobre el proyecto. Primero, se explorará la especificación de requisitos, aclarando lo que debe y no debe cumplir el proyecto. Además, se analizarán las especificaciones del sistema, donde se incluirá un análisis de casos de uso, una propuesta de solución al problema y se generarán los casos de uso. Por último, se realizará un modelo del dominio.

## 3.1. Especificación de requisitos

Necesitamos comprender y documentar las necesidades del sistema y las expectativas del mismo respecto al cliente-tutor. Esto proporcionará la base para el diseño, implementación y evaluación exitoso del software. A lo largo de esta sección se definirán los requisitos funcionales y no funcionales.

### 3.1.1. Requisitos funcionales

Dentro de esta subsección se describirán los requisitos de funcionalidad del sistema, dejando claro las acciones y capacidades esperadas del mismo.

- Usuario debe poder visualizar información obtenida tras consulta
- Usuario debe poder exportar los resultados tras la búsqueda ya sea por artículo, autor o revista en diferentes formatos como XLSX y PDF

- Permitir al usuario poder consultar información alojada en el conjunto de datos a través de búsquedas
- Permitir al usuario filtrar información de artículos, según los datos relevantes de la búsqueda
- Usuario debe obtener un mensaje de error si no encontrara resultados en los conjuntos de datos similares a la consulta
- Aportar a la interfaz gráfica accesibilidad
- El sistema debe ser capaz de mostrar indicadores de calidad, número total de citas, índice-h, promedio de citas anuales, promedio de FWCI de cada autor
- Usuario debe poder ver análisis de los indicadores de calidad u otros datos, mediante gráficas u herramientas visuales que faciliten compresión
- El sistema debe ser capaz de exportar los gráficos generados

### 3.1.2. Requisitos no funcionales

Se redactarán en este apartado los aspectos que están relacionados con el grado de cumplimiento de los requisitos funcionales.

- Interfaz gráfica intuitiva a la hora de llevar a cabo búsquedas, descargas de datos y generación de gráficos.
- Interfaz gráfica debe ser fácil de lanzar, sin necesidad de configuración previa.
- Paquete debe ser modulable, es decir, permitir facilidad a la hora de incorporar un módulo.
- El sistema debe de ser preciso a la hora de realizar gráficos y obtener datos, para garantizar fiabilidad de la información.
- El sistema debe ser capaz de manejar grandes volúmenes de datos de manera eficiente, para dar una respuesta rápida al usuario.

### 3.1.3. Diagrama de caso de uso

Una vez especificados los requisitos que tendrá nuestro proyecto, debemos descomponer estos en pequeñas partes del proyecto y donde cada parte cumplirá un

requisito específico. En la siguiente figura 3.1, se mostrarán las iteraciones entre el usuario y nuestro sistema de análisis de información bibliométrica desde diversas APIs (**BiblioMetrics**).

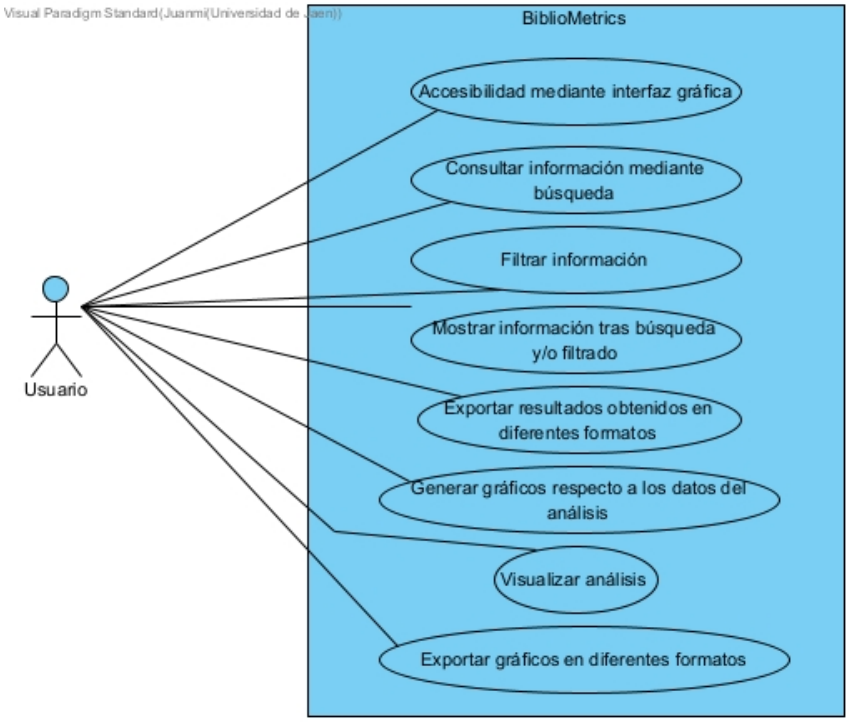



Figura 3.1: Diagrama de caso de uso de BiblioMetrics
**Fuente: Elaboración propia.**

### 3.1.4. Casos de uso

Un caso de uso representa una unidad funcional de un usuario o más (actores) con un sistema. Ahora se desglosarán los casos de uso que engloban a este proyecto, explicando los items mínimos que debe cumplir.

| Código caso de uso | Caso de uso |
|---|---|
| CU-01 | Accesibilidad mediante interfaz gráfica |
| CU-02 | Consultar artículos mediante búsquedas |
| CU-03 | Filtrar información |
| CU-04 | Mostrar información obtenida tras búsqueda y/o filtrado |
| CU-05 | Exportar resultados obtenidos en diferentes formatos |
| CU-06 | Visualizar análisis de indicadores de calidad |
| CU-07 | Generar gráficos respecto a los datos del análisis |
| CU-08 | Exportar gráficos del análisis |

Tabla. 3.1: Listado de casos de uso

| **CU-01** | **Accesibilidad mediante interfaz gráfica** |
|---|---|
| Actores | Usuario |
| Pre-condiciones | Usuario debe haber instalado el paquete **"BiblioMetrics"** junto a todas las dependencias necesarias. |
| Post-condiciones | Ninguna |
| Escenario principal | 1. Usuario ejecuta el comando `runGUI()`, pasando como parámetro el correo corporativo y la contraseña<br>2. Interfaz gráfica se carga correctamente<br>3. Usuario puede acceder a todas las funcionalidades de la aplicación |
| Escenarios alternativos | Ninguno. |
| Excepciones | Si la interfaz gráfica no se carga correctamente, el usuario puede intentar recargar la aplicación. |

Tabla. 3.2: Accesibilidad mediante interfaz gráfica

| **CU-02** | **Consultar artículos mediante búsquedas** |
|---|---|
| Actores | Usuario |
| Pre-condiciones | APIs estén disponibles y que previamente se haya ejecutado el comando `runGUI()` |
| Post-condiciones | Usuario recibe resultados basados en su consulta. |
| Escenario principal | 1. Usuario especifica lo que desea buscar en el apartado dispuesto para eso.<br>2. La aplicación procesa la consulta y busca en la API o APIs elegidas. |
| Escenarios alternativos | 1. Usuario ejecuta el comando `search()` del módulo de la API elegida, pasando los parámetros necesarios<br>2. Se procesa la consulta y busca en la API elegidas |
| Excepciones | Si la API no está disponible o bien no encuentra resultados, se mostrará un mensaje de error al usuario. |

Tabla. 3.3: Consultar artículos mediante búsquedas

| **CU-03** | **Filtrar información** |
|---|---|
| Actores | Usuario |
| Pre-condiciones | Usuario realiza una consulta y se muestran los resultados |
| Post-condiciones | Usuario ve resultados filtrados según criterios seleccionados |
| Escenario principal | 1. Usuario comprueba resultados de la consulta<br>2. Especifica los criterios de filtrado que quiere<br>3. Aplica los filtros<br>4. Los resultados se actualizan para mostrar solo los datos que coinciden con los filtros seleccionados. |
| Escenarios alternativos | 1. Usuario ejecuta el comando `search()`, añadiendo los filtros elegidos por el usuario.<br>2. Los resultados de la consulta será la búsqueda con los requisitos elegidos por usuario. |
| Excepciones | Si no se encuentran resultados tras aplicar los requisitos, se notifica la usuario. |

Tabla. 3.4: Filtrar información

| **CU-04** | **Mostrar información obtenida tras búsqueda y/o filtrado** |
|---|---|
| Actores | Usuario |
| Pre-condiciones | El usuario formula una consulta y, si es conveniente, puede aplicar o no filtros |
| Post-condiciones | El usuario visualizará resultados precisos que coincidan con los filtros seleccionados. |
| Escenario principal | 1. Aplicación muestra resultados en la interfaz gráfica<br>2. Usuario puede navegar y explorar resultados para obtener más información |
| Escenarios alternativos | Si la consulta es producida por la línea de comandos, los resultados se mostrarán impresos en la línea de comandos. |
| Excepciones | Si los resultados no se muestran debido a problemas con la conexión con las APIs, se notificará al usuario. |

Tabla. 3.5: Mostrar información obtenida tras búsqueda y/o filtrado

| **CU-05** | **Exportar resultados obtenidos en diferentes formatos** |
|---|---|
| Actores | Usuario |
| Pre-condiciones | Usuario realiza una consulta, la cual se muestran los resultados. |
| Post-condiciones | Usuario obtiene resultados en formato deseado |
| Escenario principal | 1. Tras analizar resultados obtenidos en la consulta, usuario pulsa en opción exportar<br>2. Se muestran opciones para obtener los resultados en diferentes formatos<br>3. Usuario elige el formato y confirma<br>4. Aplicación genera y descarga archivo en formato elegido por el usuario |
| Escenarios alternativos | Ninguno. |
| Excepciones | Si al generar el archivo, se produce algún error, mostrar un mensaje de error, que indique que se intente más tarde. |

Tabla. 3.6: Exportar resultados obtenidos en diferentes formatos

| **CU-06** | **Visualizar análisis de indicadores de calidad** |
|---|---|
| Actores | Usuario |
| Pre-condiciones | El usuario ha realizado una consulta y se obtienen resultados y el usuario selecciona un artículo a analizar. |
| Post-condiciones<br>Escenario principal | El usuario visualiza los datos del análisis del artículo elegido.<br>1. Tras elegir artículo, autor o revista, el usuario debe hacer clic sobre la opción análisis<br>2. El usuario puede investigar los resultados para obtener más detalles |
| Escenarios alternativos | Ninguno. |
| Excepciones | Si hubiese un error al obtener los datos, se mostraría un mensaje de error. |

Tabla. 3.7: Visualizar análisis de indicadores de calidad

| **CU-07** | **Generar gráficos respecto a los datos del análisis** |
|---|---|
| Actores | Usuario |
| Pre-condiciones | El usuario ha obtenido el análisis de uno de los artículos. |
| Post-condiciones | El usuario visualiza los datos del análisis del artículo elegido representados gráficamente |
| Escenario principal | 1. Tras obtener el análisis, el usuario selecciona la opción análisis gráfico<br>2. La aplicación genera el/los gráfico/s correspondiente/s utilizando los datos obtenidos<br>3. Usuario obtiene el gráfico en la interfaz gráfica |
| Escenarios alternativos | Ninguno. |
| Excepciones | Si hubiese algún error al generar el gráfico o obtener datos, se muestra un mensaje de error. |

Tabla. 3.8: Generar gráficos respecto a los datos del análisis

| **CU-08** | **Exportar gráficos de análisis** |
|---|---|
| Actores | Usuario |
| Pre-condiciones | Usuario ha obtenido los resultados del análisis. |
| Post-condiciones | Usuario puede descargar los resultados del análisis en el formato que desee. |
| Escenario principal | 1. Usuario selecciona la opción exportar<br>2. Aplicación genera y descarga el archivo. |
| Escenarios alternativos | Ninguno. |
| Excepciones | Si hubiese un error al exportar, en el formato de exportación, en la ruta, se muestra un mensaje de error. |

Tabla. 3.9: Exportar gráficos del análisis

## 3.2. Modelo del dominio

Una vez que se alcanzan los requisitos que necesita el cliente y se han obtenido los casos de uso correspondientes, resulta crucial diseñar un modelo del dominio respecto a estos casos de uso. Este modelo sirve como diagrama conceptual que captura la idea del proyecto a abordar. En esencia, identifica las entidades más representativas del proyecto y muestra claramente la relaciones entre estos conceptos, sin aún representar objetos software. Además, representa la interacción que puede necesitar el usuario respecto a nuestro proyecto. Este proceso establecerá unas bases sólidas para el diseño e implementación del proyecto.

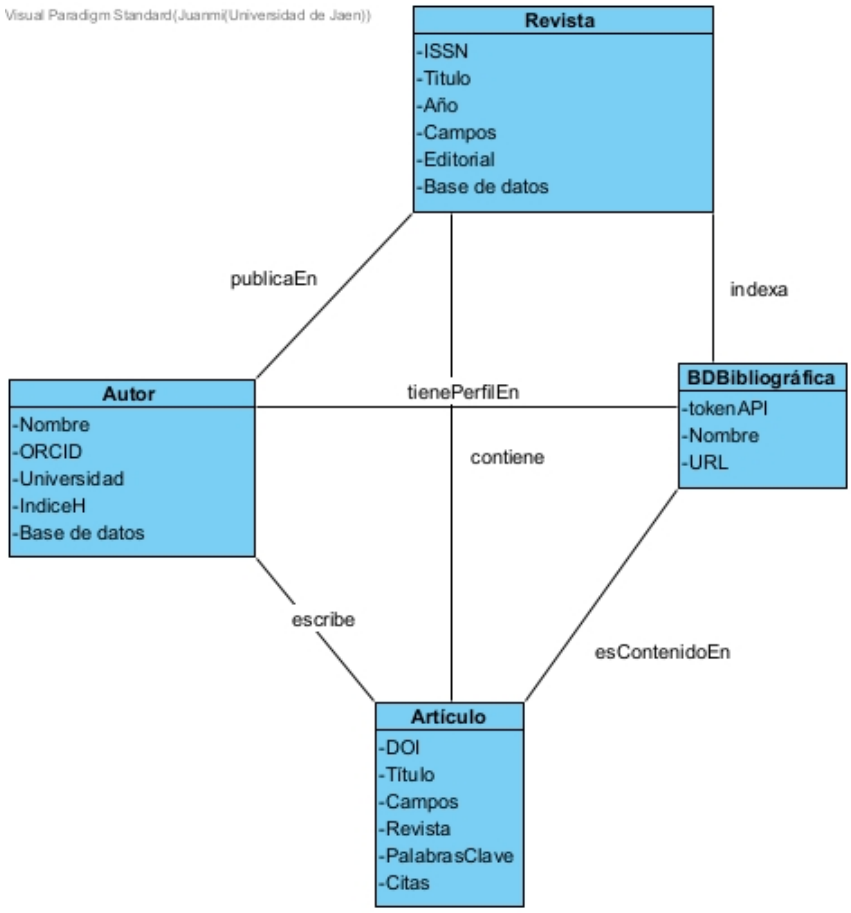



Figura 3.2: Diagrama de Modelo del dominio
**Fuente: Elaboración propia.**

En el diagrama 3.2 se presenta la estructura de diseño del proyecto. El esquema muestra una base de datos bibliográfica donde un artículo, que constituye la unidad mínima de información, puede estar escrito por uno o varios autores. A su vez, el artículo está publicado en una revista. Además, el autor tiene la posibilidad de publicar múltiples artículos en diversas revistas. Asimismo, el autor cuenta con un perfil en una base de datos bibliométrica, dado que en esta base de datos se encuentran indexados

uno o más de sus artículos. Por último, las bases de datos bibliográficas se encargan de indexar revistas que contienen dichos artículos.

# Capítulo 4

# PLANIFICACIÓN

En este apartado del documento, se procederá a detallar la división del proyecto en tareas o elementos. También se establecerá la frecuencia y duración entre las iteraciones entre *sprints*. Además, se realizará un plan claro y estructurado desde su inicio a su finalización. Para así realizar una gestión eficiente del tiempo y de los recursos disponibles.

La reunión entre sprint con el cliente-tutor se llevará a cabo cada vez que necesite de él. Respecto a la fecha de entrega, se tuvo que modificar a la estimación temporal (2.3), debido a la imposibilidad de dedicarle una jornada completa de 8 horas diarias, pasando a ser unas 4 horas diarias. Dejando como fecha final, el **6 de septiembre**. Dando margen de error en caso de surgir excepciones inesperadas. Al modificar la fecha de entrega, el plazo será de 17 semanas, lo que implica la realización de 17 sprint semanales.

En la siguiente tabla 4.1, se detalla la planificación por semanas para el desarrollo del proyecto, organizados por *sprints*, con las tareas correspondientes previstas para cada uno.

## 4.1. Gestión Temporal

| Sprint | Fecha inicio | Fecha final | Tareas |
|---|---|---|---|
| **1** | 07/05/2024 | 14/05/2024 | ■ Ejecución del diseño del proyecto<br>■ Crear repositorio de código en GitHub<br>■ Configurar entorno de desarrollo en RStudio |
| **2** | 15/05/2024 | 21/05/2024 | ■ Investigación de paquetes respecto al proyecto<br>■ Ejecución del diseño del proyecto<br>■ Inicio y adaptación en el uso del lenguaje R y sus paquetes |
| **3** | 22/05/2024 | 28/05/2024 | ■ Inicio y adaptación en el uso del lenguaje R y sus paquetes<br>■ Exploración de las diferentes APIs disponibles<br>■ Implementar interfaz gráfica básica |
| **4** | 29/05/2024 | 04/06/2024 | ■ Exploración de las diferentes APIs disponibles<br>■ Implementar interfaz gráfica básica<br>■ Implementar en interfaz gráfica resultados de consulta |

| Sprint | Fecha inicio | Fecha final | Tareas |
|---|---|---|---|
| **5** | 05/06/2024 | 11/06/2024 | ▪ Exploración de las diferentes APIs disponibles<br>▪ Implementar interfaz gráfica básica<br>▪ Implementar en interfaz gráfica resultados de consulta |
| **6** | 12/06/2024 | 18/06/2024 | ▪ Exploración de las diferentes APIs disponibles<br>▪ Implementar interfaz gráfica resultados de consulta<br>▪ Implementar en interfaz gráfica análisis de consulta |
| **7** | 19/06/2024 | 25/06/2024 | ▪ Exploración de las diferentes APIs disponibles<br>▪ Implementar interfaz gráfica resultados de consulta<br>▪ Implementar en interfaz gráfica análisis de consulta |
| **8** | 26/06/2024 | 02/07/2024 | ▪ Exploración de las diferentes APIs disponibles<br>▪ Implementar en interfaz gráfica análisis de consulta<br>▪ Realizar consulta de datos de Artículos |

| Sprint | Fecha inicio | Fecha final | Tareas |
|---|---|---|---|
| **9** | 02/07/2024 | 09/07/2024 | ▪ Exploración de las diferentes APIs disponibles<br>▪ Implementar en interfaz gráfica análisis de consulta<br>▪ Realizar consulta de datos de Artículos<br>▪ Realizar consulta de datos de Autor |
| **10** | 09/07/2024 | 16/07/2024 | ▪ Realizar consulta de datos de Revistas<br>▪ Realizar consulta de datos de Autor<br>▪ Desarrollar la funcionalidad de representar resultados |
| **11** | 16/07/2024 | 23/07/2024 | ▪ Investigar paquetes sobre generación de gráficos en R<br>▪ Integrar funcionalidad para obtener datos para análisis Artículos |
| **12** | 23/07/2024 | 30/07/2024 | ▪ Integrar funcionalidad para obtener datos para análisis Autor<br>▪ Integrar funcionalidad para obtener datos para análisis Revista<br>▪ Desarrollar métodos para generar y mostrar gráficos |

| Sprint | Fecha inicio | Fecha final | Tareas |
|---|---|---|---|
| **13** | 30/07/2024 | 06/08/2024 | ▪ Desarrollar métodos para generar y mostrar gráficos |
| **14** | 06/08/2024 | 13/08/2024 | ▪ Consultar descarga de gráficos obtenidos tras análisis<br>▪ Implementar exportación en diversos formatos |
| **15** | 13/08/2024 | 20/08/2024 | ▪ Consultar descarga de datos obtenidos tras consulta<br>▪ Implementar exportación en diversos formatos |
| **16** | 20/08/2024 | 27/08/2024 | ▪ Generación de documentación del paquete<br>▪ Filtrado de datos obtenidos desde las diferentes APIs<br>▪ Conclusión de proyecto y revisión de documentación |
| **17** | 27/08/2024 | 06/09/2024 | ▪ Conclusión de proyecto y revisión de documentación |

Tabla. 4.1: Planificación temporal detallada de tareas

A medida que se efectúe cada tarea, ese reflejará correctamente en la memoria del proyecto. Esto dejará claro, qué se ha llevado a cabo en la tarea y por qué se hizo de la manera que se especifica, siguiendo un orden.

### 4.1.1. Diagrama de Gantt

Este diagrama proporcionará una representación visual detallada de la estimación temporal del proyecto, facilitando la organización de las tareas en función de su duración y orden de ejecución. De esta manera, será posible planificar de manera efectiva las actividades y establecer hitos clave. Además, permitirá llevar a cabo un seguimiento continuo del progreso, identificando posibles retrasos o desviaciones del plan original, lo que contribuirá a la toma de decisiones correctivas en caso necesario.

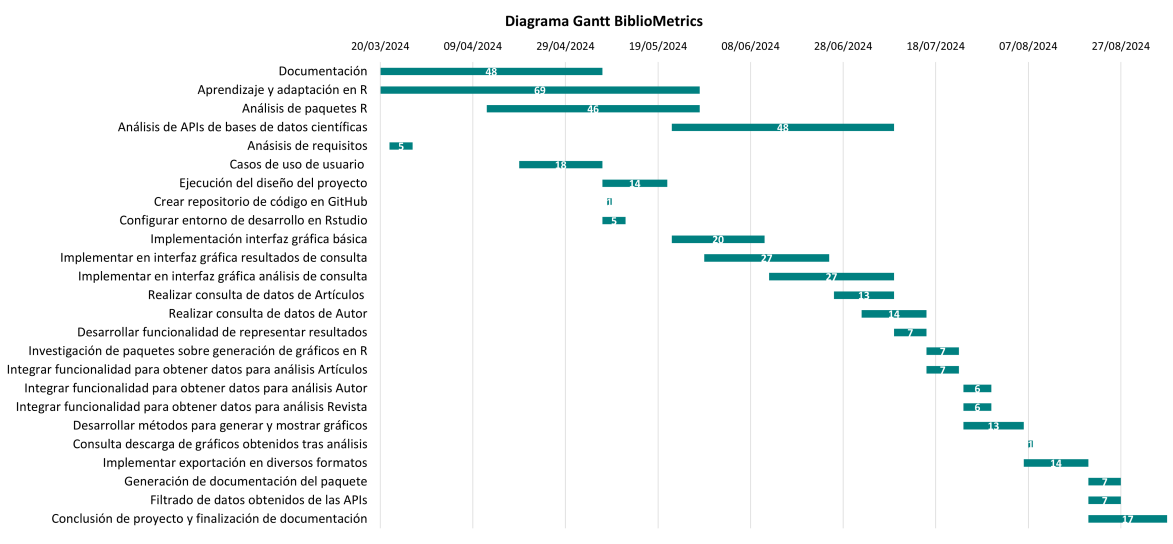



Figura 4.1: Diagrama de Gantt de BiblioMetrics
**Fuente: Elaboración propia.**

# Capítulo 5

# DISEÑO

Este apartado es uno de los más críticos del proyecto, ya que proporciona una descripción detallada y estructurada de lo analizado y planificado. En esencia, ofrece una visión clara de cómo se abordará el desarrollo del proyecto, las consideraciones tenidas en cuenta y la estructura del software para cumplir con los requisitos establecidos por el cliente. Por tanto, resulta fundamental para comprender el enfoque aplicado y garantizar que el diseño del software siga alineado con los objetivos del proyecto. En consecuencia, en esta sección se ofrecerá una explicación del sistema, comenzando desde los componentes más abstractos hasta los más concretos.

## 5.1. Diagrama de sistema

El diagrama 5.1 proporciona una representación visual que permite comprender de un vistazo la organización del proyecto. Presenta una vista general de alto nivel, mostrando cada uno de los elementos que constituyen el sistema en su totalidad.

## 5.2. Diagramas de clases

Como se explicó en el apartado anterior, el diseño seguirá un enfoque descendente en el nivel de abstracción. Por lo tanto, se procederá a obtener el diseño de clases y métodos. En el lenguaje de programación escogido para este proyecto, no utiliza clases como otros lenguajes orientados a objetos. Por tanto, este diagrama describe

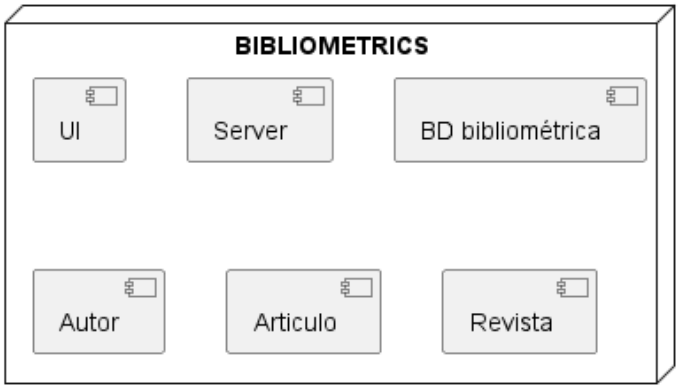



Figura 5.1: Diagrama de sistema
**Fuente: Elaboración propia.**

el sistema, mostrando la simulación de clases, atributos y las relaciones entre ellas, pero realmente no será así a la hora de la ejecución en el lenguaje de programación.

Para explicar el diagrama 5.2. Lo principal es entender las clases en las cuales se ha dividido este diagrama.

- Clase `UI`: esta clase llevará a cabo todas las tareas de implementación de la interfaz gráfica de la parte estática, incluyendo la generación de cada pantalla. El comportamiento de esta clase, está completamente unido a la clase `Server`, que hará la función de servidor de la aplicación
- Clase `Server`: gracias a sus métodos, realiza el funcionamiento no visible de la aplicación. Obtiene datos de las bases de datos bibliométricas, realiza gráficos. Además, también genera partes de la aplicación dinámica
- Clase `BD Bibliométrica`: representa la entidad más importante del proyecto, ya que es de donde obtenemos todos los datos necesarios por la aplicación. Contienen atributos que usaremos a la hora de la implementación, para acceder a las APIs de cada base de datos
- Clases `Revista`, `Autor` y `Artículo`: proporcionan a la aplicación los atributos que serán representados tras la búsqueda y métodos necesarios para analizar los datos obtenidos. Posteriormente estos datos serán pasados a `Server`

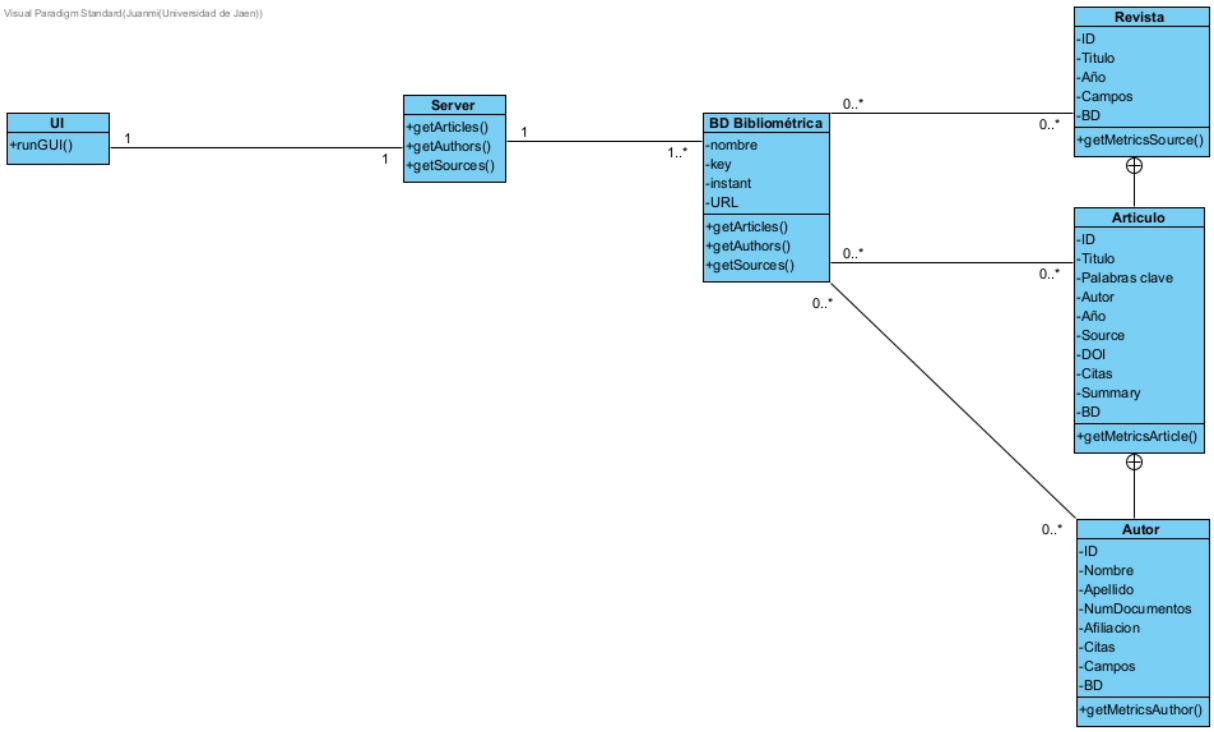



Figura 5.2: Diagrama de clases
**Fuente: Elaboración propia.**

## 5.3. Diagramas de secuencia

Los diagramas de secuencia se utilizan para representar la interacción entre objetos que se produce en un caso de uso o para operaciones concretas [8], a continuación se presentan diagramas relacionados con los casos de uso y algunas operaciones que serán interesantes en el proyecto.

| Apartado | Diagrama de secuencia |
|---|---|
| 5.3 | Diagrama de secuencia descarga e instalación de paquete |
| 5.4 | Diagrama de secuencia de acceso a la interfaz gráfica |
| 5.5 | Diagrama de secuencia de consultas de datos |
| 5.6 | Diagrama de secuencia de visualización de información tras consulta |
| 5.7 | Diagrama de secuencia de filtrado de resultados |
| 5.8 | Diagrama de secuencia de exportado de resultados en diferentes formatos |
| 5.9 | Diagrama de secuencia de generación de análisis |

Tabla. 5.1: Listado de diagramas de secuencia

### 5.3.1. Descarga e instalación del paquete

En este subapartado se presentará el siguiente diagrama 5.3 de secuencia para la descarga e instalación del paquete. Aunque no se trata de un caso de uso, es una operación esencial que debe llevarse a cabo para obtener el paquete. Por lo tanto, es fundamental que esta operación quede clara para el usuario.

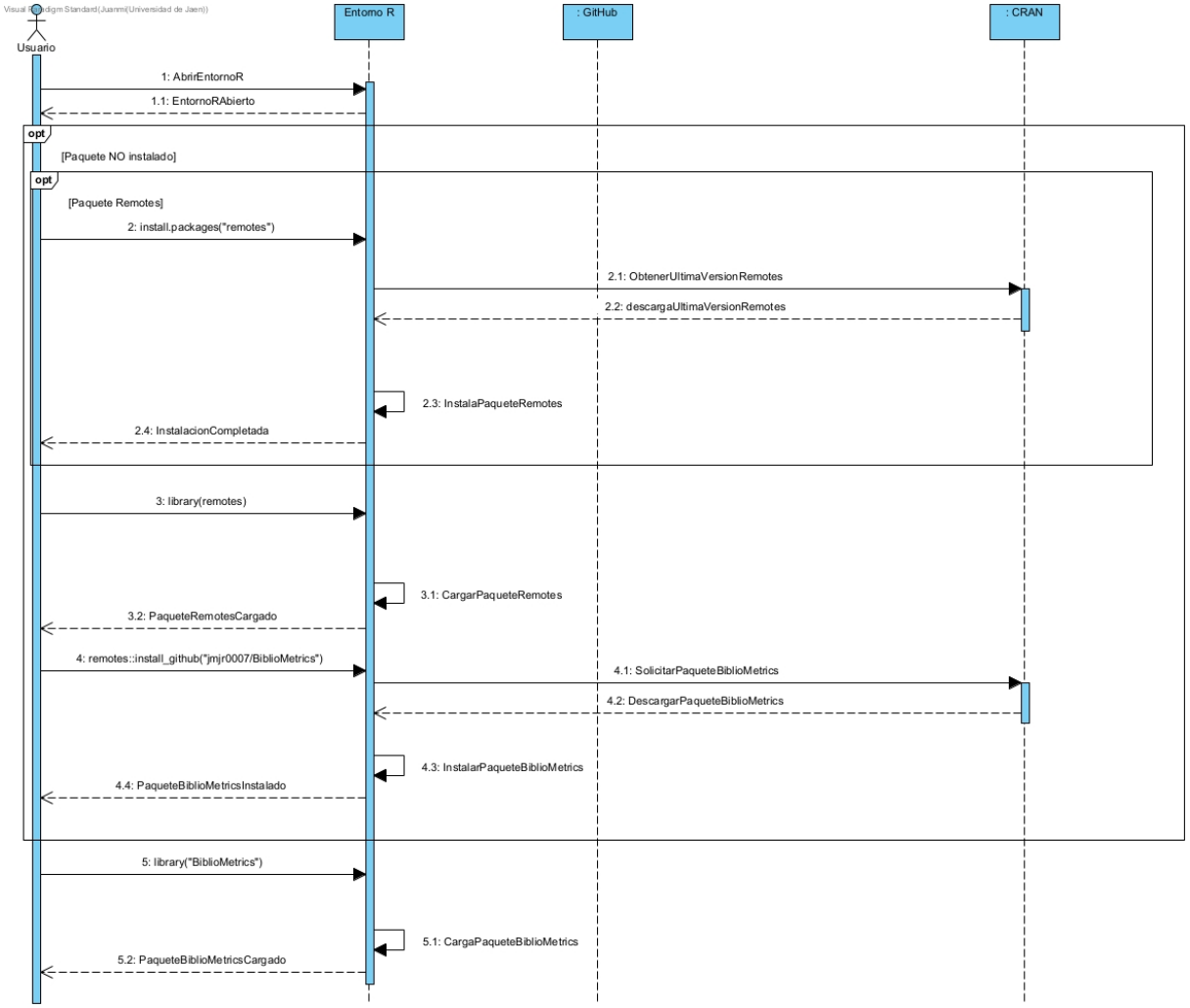



Figura 5.3: Diagrama de secuencia descarga e instalación de paquete
**Fuente: Elaboración propia.**

### 5.3.2. Acceso a la interfaz gráfica

En este diagrama mostraremos cómo el usuario interactuará para llevar a cabo el acceso a la interfaz gráfica.

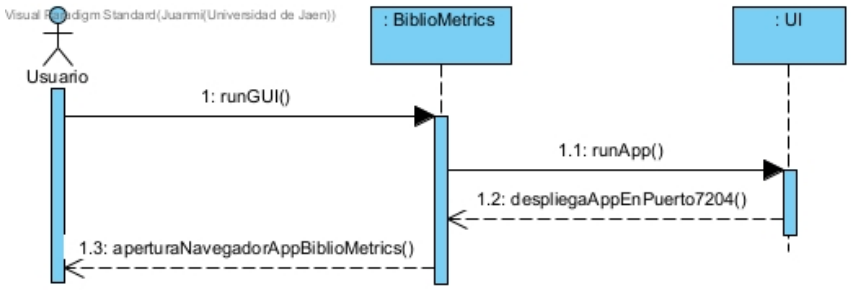



Figura 5.4: Diagrama de secuencia de acceso a la interfaz gráfica
**Fuente: Elaboración propia.**

### 5.3.3. Consulta de datos

En el diagrama de este caso de uso, es necesaria la intervención de cada una de las posibles APIs, desde las cuales se obtendrán los datos para ser mostrados al usuario, cumpliendo con los requisitos de su búsqueda.

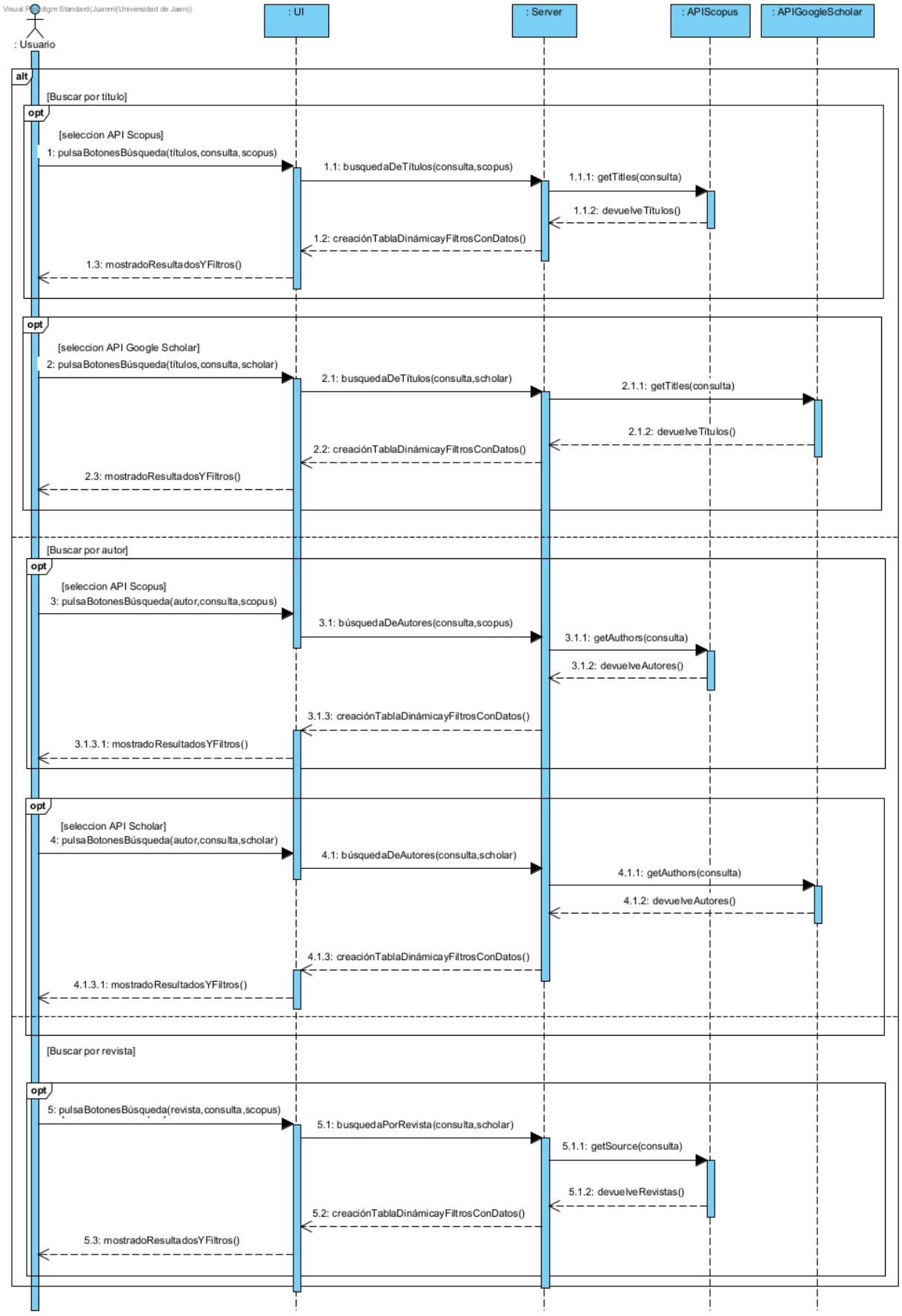



Figura 5.5: Diagrama de secuencia de consultas de datos
**Fuente: Elaboración propia.**

### 5.3.4. Visualización de información tras consulta

En el caso de la visualización de información después de la consulta, se hará referencia a métodos de diagramas de secuencia anteriores, de acceso a la interfaz gráfica 5.4 y de consultas de datos 5.5. Por lo tanto, no será necesario que intervengan las APIs en este proceso, ya que el paquete contendrá la información. Esta información se pasará al servidor, que se encargará de generar la tabla dinámica, la cual se actualizará según los filtros seleccionados. Por lo tanto, cada vez que se obtengan datos, el servidor deberá reiniciar los datos de los filtros y eliminar aquellos que no sean necesarios. Finalmente, la aplicación mostrará tanto los datos como los filtros al usuario.

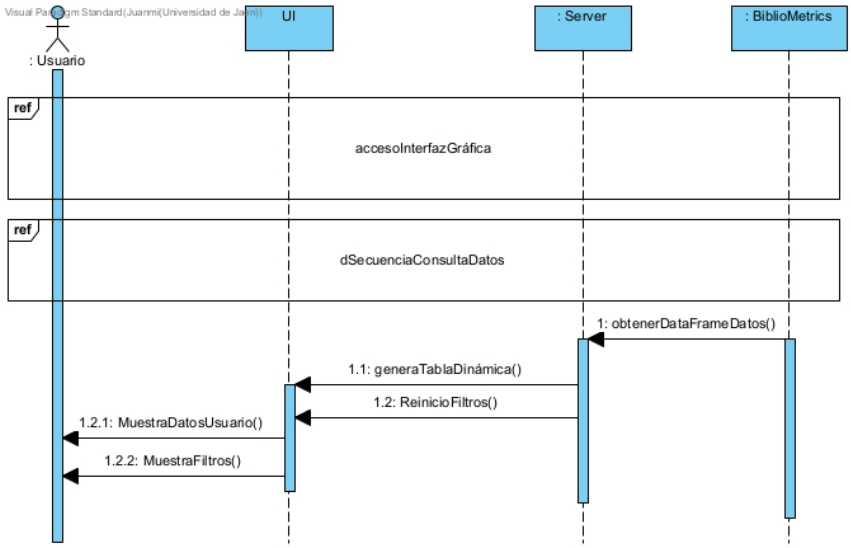



Figura 5.6: Diagrama de secuencia de visualización de información tras consulta
**Fuente: Elaboración propia.**

### 5.3.5. Filtrado de resultados

En este caso de uso, se procederá a filtrar los datos obtenidos en el diagrama de secuencia de visualización de información 5.6. Por lo tanto, no será necesaria la intervención de las APIs en este proceso. Los datos descargados, que ya están en la clase `Server`, serán los que se utilicen para los filtros. Tras mostrar al usuario la tabla con sus respectivos filtros, este seleccionará alguno de ellos y, de forma dinámica, la tabla mostrará exclusivamente aquellos registros que coincidan con los requisitos del usuario.

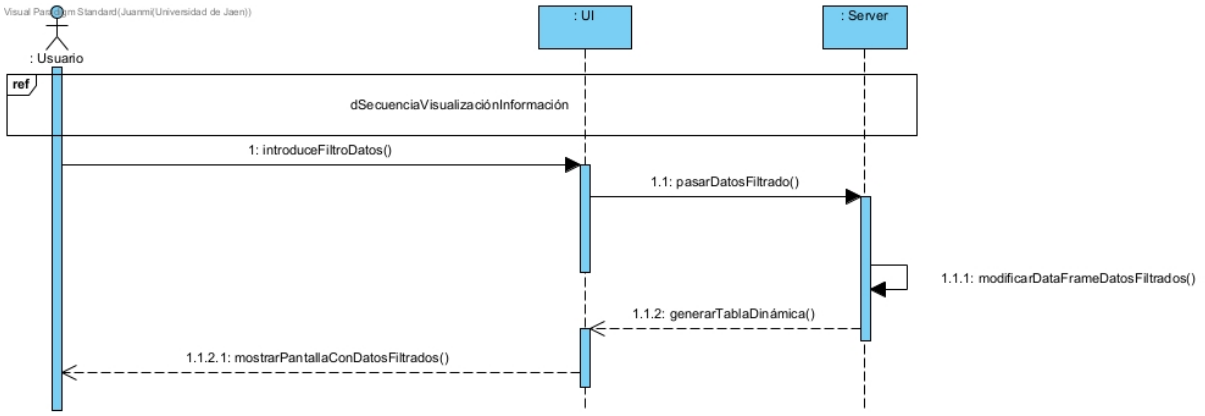



Figura 5.7: Diagrama de secuencia de filtrado de resultados
**Fuente: Elaboración propia.**

### 5.3.6. Exportado de resultados en diferentes formatos

Este diagrama de secuencia muestra cómo el usuario debe llevar a cabo la exportación de los datos obtenidos. En este caso de uso, no será necesaria la intervención de las APIs seleccionadas, ya que los datos estarán en la clase `Server` tras el diagrama de secuencia de filtrado de resultados 5.7, hayan sido filtrados o no. Estos datos se exportarán en formato PDF o Excel, dependiendo de la elección del usuario.

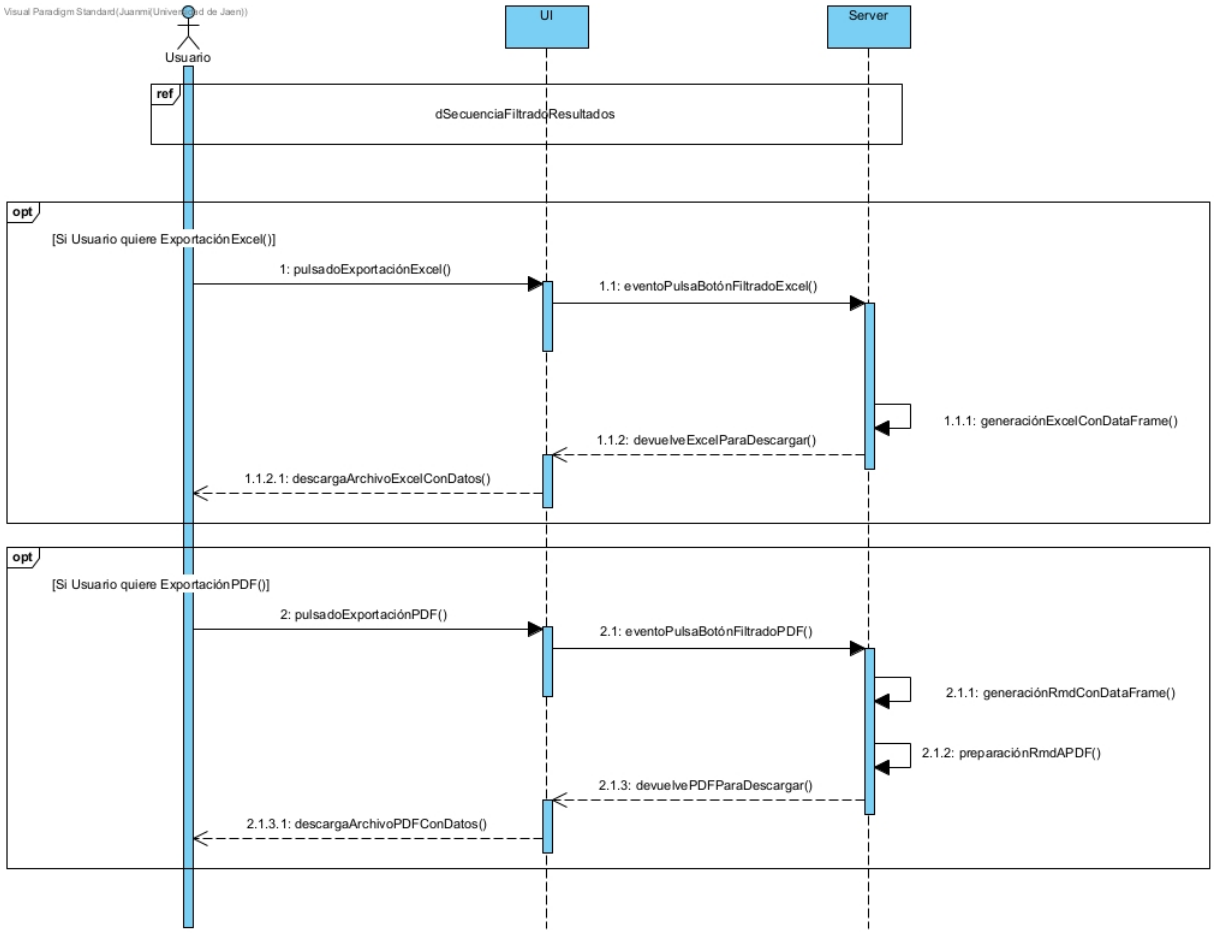



Figura 5.8: Diagrama de secuencia de exportado de resultados en diferentes formatos
**Fuente: Elaboración propia.**

### 5.3.7. Generación de análisis

Este diagrama ilustra cómo el usuario solicita una visualización gráfica de los datos obtenidos tras realizar una consulta previa.

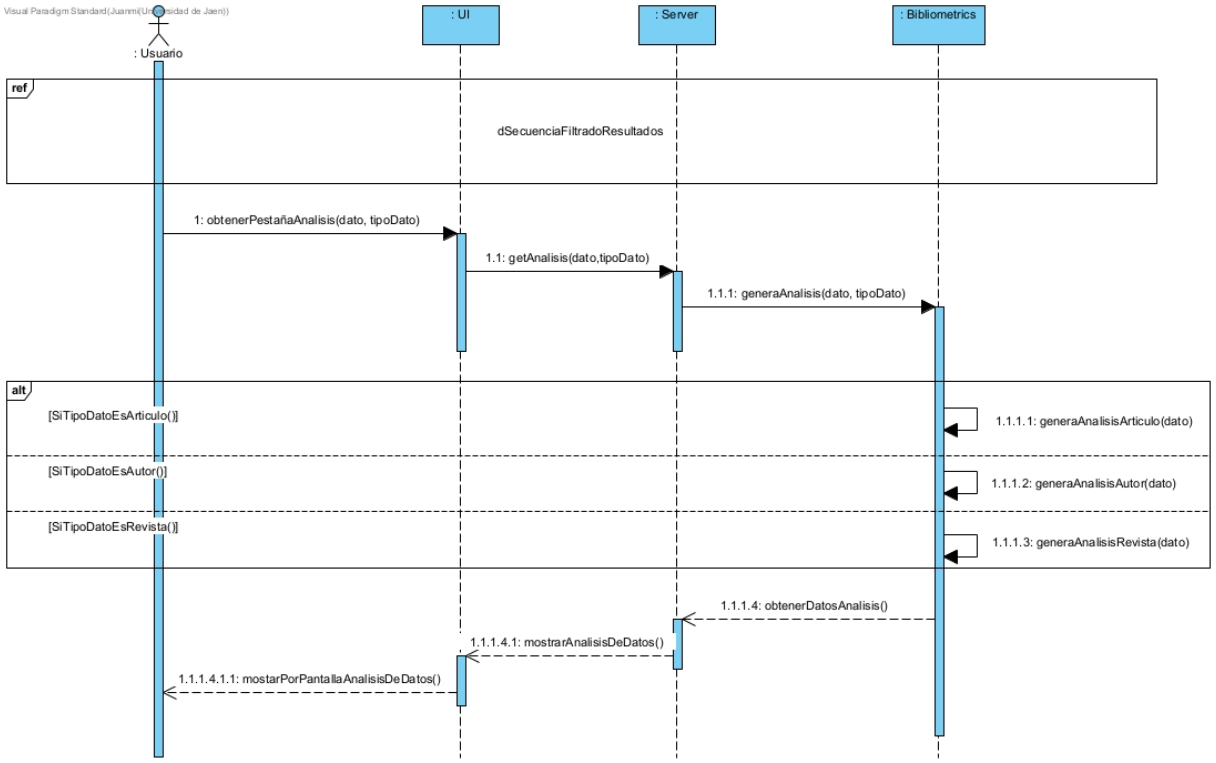



Figura 5.9: Diagrama de secuencia de generación de análisis
**Fuente: Elaboración propia.**

# Capítulo 6

# MANUAL DE ESTILO

En este capítulo se recogen las normas generales de estilo visual que va a seguir la aplicación desarrollada en el proyecto. Así se conseguirá tener presencia digital y los usuarios de la aplicación reconocerán el proyecto de manera rápida. Además, también en esta sección, se aclara la distribución de los elementos, que hará de la aplicación ser intuitiva, limpia y sencilla.

## 6.1. Bocetos

La primera etapa del estilo y diseño de la interfaz gráfica, es obtener una representación visual y esquemática, es decir, realizar un prototipo o diseño. El propósito de estos no es otro que diseñar los pilares de la aplicación, colocando las estructuras más grandes y dejando claro las posiciones de los elementos más básicos.

Como se mencionó anteriormente, se busca que la aplicación sea lo más intuitiva posible y accesible para cualquier usuario. Por ello, se ha diseñado una interfaz sin ventanas emergentes, concentrando todas las funcionalidades en la pantalla principal (6.1). Esta pantalla, de diseño sencillo, contará con un área dedicada exclusivamente a las tareas de búsqueda. En esta sección principal, se proporcionarán instrucciones claras y concisas para guiar al usuario durante el proceso de búsqueda y análisis.

Después de realizar la búsqueda (6.2), el panel principal, donde se mostraban las instrucciones, cambiará para mostrar los resultados de la búsqueda. Además, aparecerá un bloque a la izquierda con filtros que permitirán al usuario refinar su búsqueda según sus necesidades.

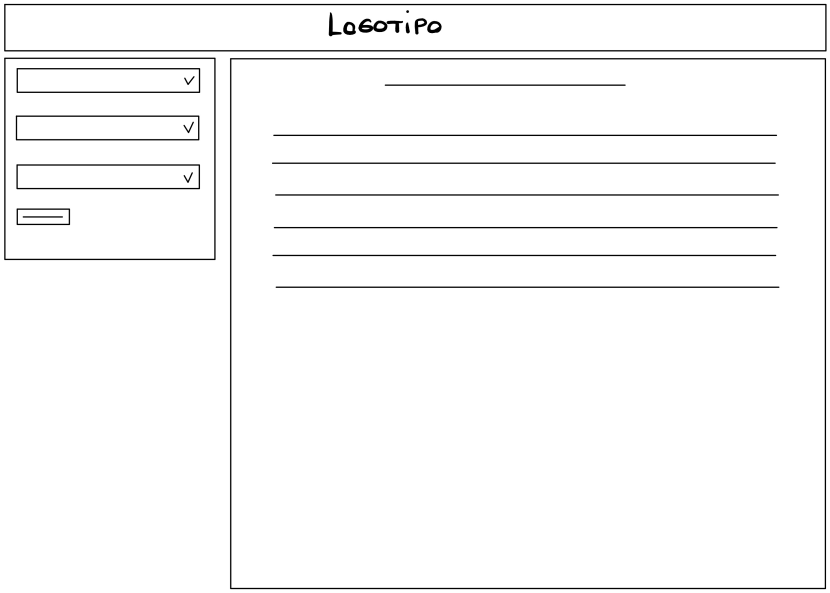



Figura 6.1: Boceto de pantalla principal
**Fuente: Elaboración propia.**

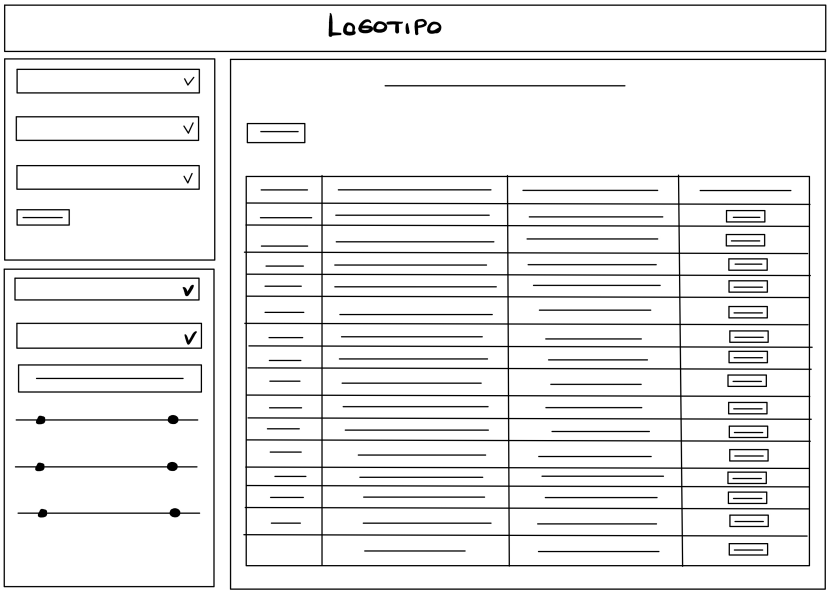



Figura 6.2: Boceto de pantalla inicial tras una búsqueda
**Fuente: Elaboración propia.**

En la tabla de resultados, en la parte derecha, se mostrará un botón que permitirá iniciar el análisis. Al pulsar dicho botón, se abrirá una pestaña de análisis (6.3), la cual mostrará el contenido correspondiente al hacer clic sobre ella.

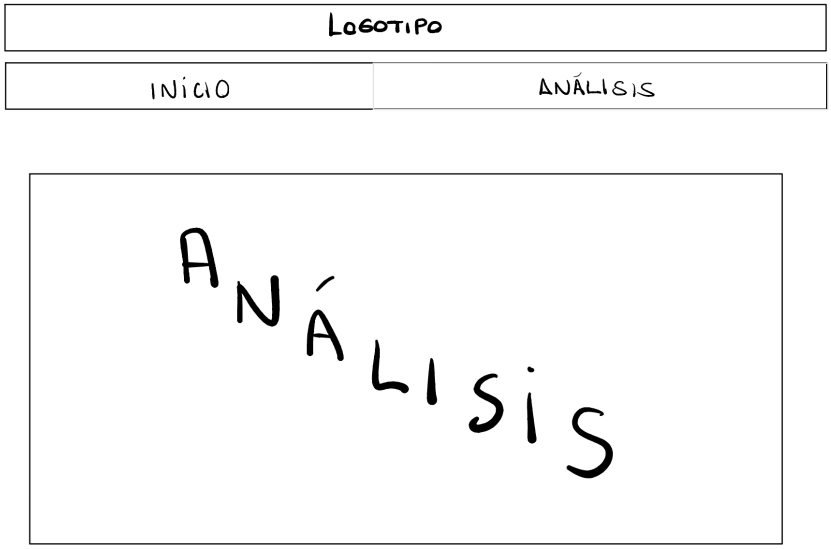



Figura 6.3: Diagrama de la pantalla de análisis
**Fuente: Elaboración propia.**

## 6.2. Mock-up

La definición pura de *Mock-up* o maquetas, no es más que “una representación más avanzada del diseño gráfico y comunicativo de un proyecto” [9].

Por tanto, se procede a realizar una representación más avanzada, con mucho más detalle y más cercana a la versión final de cómo será el diseño de las vistas más importantes.

### 6.2.1. Pantalla de inicio

Esta pantalla sigue el diseño propuesto en el boceto (6.1). Se incluyen detalles adicionales, como una barra de navegación que permite acceder a la pestaña de ”Análisis” al pulsar el botón correspondiente. También se muestra la distribución de la aplicación, incluyendo la disposición de botones, filtros y la estructura de la tabla. Además, se puede anticipar la paleta de colores que se utilizará en la interfaz final.

### 6.2.2. Pantalla de análisis

En cuanto a la pantalla de análisis (6.5), no es posible crear un *mock-up* exhaustivo, ya que cada opción de análisis mostrará diferentes pantallas, lo que dificulta el


Figura 6.4: Mock-up de pantalla de inicio
**Fuente: Elaboración propia.**

diseño de un *mock-up* global. En esta maqueta no se incluyen filtros ni buscadores, únicamente se muestra la barra de navegación para cambiar de pantalla.

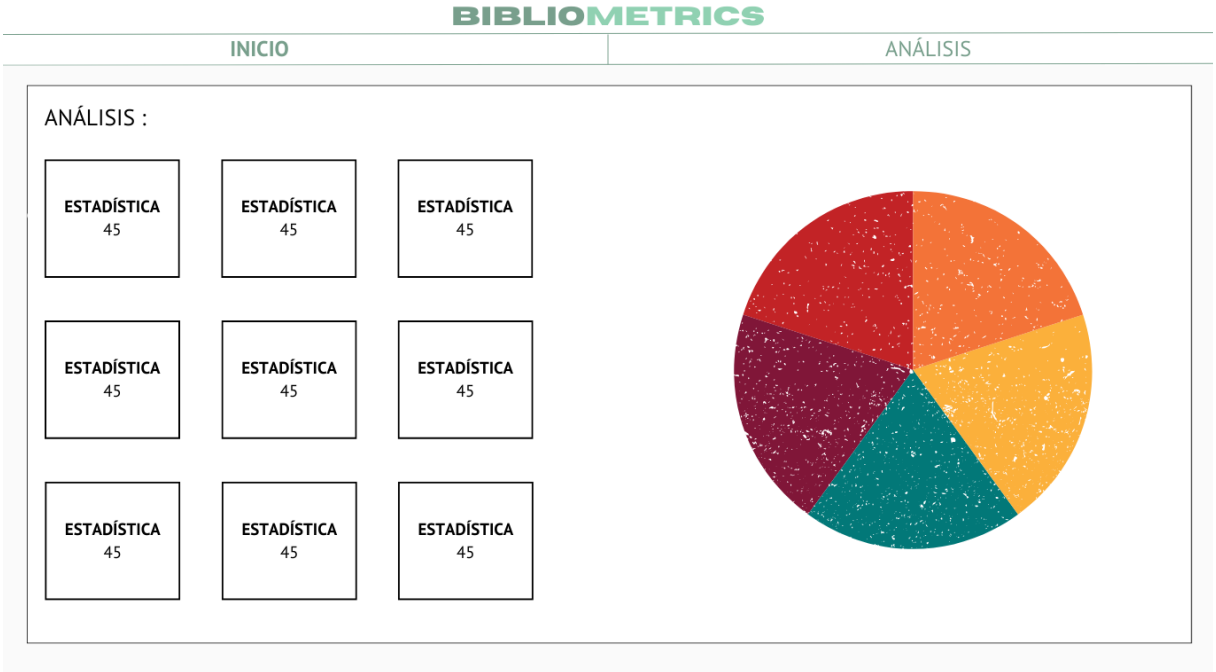



Figura 6.5: Mock-up de pantalla de análisis
**Fuente: Elaboración propia.**

## 6.3. Logotipo

La identidad visual es un elemento fundamental para atraer la atención del usuario. La elección de un logotipo sencillo responde al objetivo de mantener la sencillez y el minimalismo en toda la aplicación, en beneficio de la atracción del usuario.


Figura 6.6: Icono de la marca miniatura
**Fuente: Elaboración propia.**


Figura 6.7: Icono oficial de la marca
**Fuente: Elaboración propia.**

## 6.4. Paleta de colores

Como se puede apreciar en la figura (6.8), se utilizan diferentes tonos de verde como colores corporativos debido a que transmiten profesionalidad y confiabilidad al usuario. Además, estos tonos aseguran una buena legibilidad del texto sobre un fondo claro. Los colores seleccionados también están asociados con el conocimiento y la tecnología, conceptos centrales del proyecto.

## 6.5. Tipografías

En este apartado se debe distinguir entre las tipografías utilizadas en el proyecto y las empleadas en el logotipo de la aplicación.

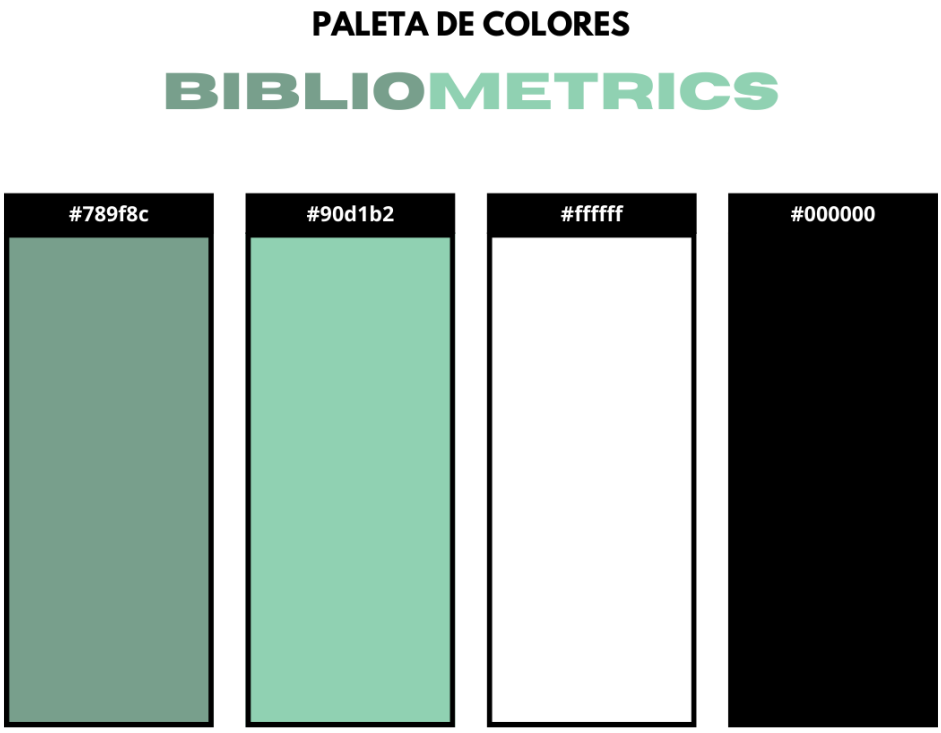



Figura 6.8: Paleta de colores BiblioMetrics
**Fuente: Elaboración propia.**

### 6.5.1. Tipografía de logotipos

A la hora de diseñar el logotipo, se utilizó la aplicación **Canva** [1]. En ella, se generaron tanto el logotipo principal de la web 6.7, en el que se usó la fuente **Horizon**, como el logotipo en formato de icono miniatura 6.6, para el cual se empleó la fuente **Canva Sans**.

### 6.5.2. Tipografía de la aplicación

En el diseño de la aplicación, es fundamental reconocer la importancia de las tipografías. Debemos seleccionar fuentes fáciles de leer para proporcionar una experiencia de usuario cómoda y agradable.

[1] Canva

La tipografía en la aplicación dependerá del sistema operativo desde el que se utilice, y será la proporcionada por la librería **Materialize** [2].

- **apple-system**: fuente de sistema de Apple, usada en macOS y iOS
- **BlinkMacSystemFont**: fuente de sistema usada por los navegadores basados en Blink (como Chrome) en macOS
- **Segoe UI**: fuente utilizada en interfaces de usuario de Windows
- **Roboto**: fuente de sistema de Android
- **Oxygen-Sans**: fuente usada en algunas distribuciones de Linux
- **Ubuntu**: fuente usada en la distribución Ubuntu de Linux
- **Cantarell**: fuente utilizada en GNOME, entorno de escritorio de algunas distribuciones de Linux
- **Helvetica Neue**: fuente utilizada en algunas versiones de macOS
- **Fuente genérica sans-serif**: utilizada como última opción si ninguna de las anteriores está disponible

[2] Framework Materialize

# Capítulo 7

# DESARROLLO

## 7.1. Tecnologías utilizadas

### 7.1.1. Herramientas de generación de diagramas

Teniendo la necesidad de encontrar una aplicación para crear los diagramas necesarios en el proyecto, optaremos por utilizar **Visual Paradigm** [1], una herramienta ya usada anteriormente en el Grado y que, por tanto se cuenta con experiencia sobre ella. Visual Paradigm ofrece herramientas visuales intuitivas y potentes para la generación de diversos tipos de diagramas. En este proyecto, requeriremos la creación de los siguientes diagramas:

- Diagrama de caso de uso 3.1.
- Diagramas de secuencia: 5.4, 5.3, 5.8, 5.7, 5.9.
- Diagrama de modelo de dominio: 3.2

Para el diagrama de sistema 5.1, no se consiguió encontrar en Visual Paradigm, se uso una extensión de Visual Studio Code, llamada PlantUML.

Además, el diagrama de Gantt 4.1, está hecho en el programa Excel, mediante sus gráficos de barras.

[1] [Web de Visual Paradigm]. (s.f.). Visual Paradigm. https://www.visual-paradigm.com/. Fecha de acceso: 25 de abril de 2024.

### 7.1.2. Herramienta para gestión de tareas y seguimientos de progreso

**Trello** [2] es una aplicación web que simplifica la gestión de proyectos y tareas de manera eficiente. Destaca por su sencillez, flexibilidad y potencia. Esta herramienta es adaptable a la preferencia del usuario. En el caso de este proyecto, aunque algunas funciones avanzadas de la aplicación no serán utilizadas, debido a que solo una persona estará a cargo de la tarea y trabajará con ella, Trello sigue siendo una elección intuitiva y óptima para organizar las tareas derivadas de las historias de usuario. Su interfaz amigable permite gestionar las tareas sin complicaciones.

### 7.1.3. Lenguajes

El lenguaje de programación utilizado como base para el desarrollo de este proyecto es **R** [3], el cual se distribuye bajo una licencia de software libre. Este lenguaje ha ganado una amplia aceptación en la comunidad científica y académica, destacándose especialmente en el ámbito del análisis de datos y la estadística. Asimismo, R ofrece una extensa variedad de herramientas gráficas que facilitan la visualización de datos de manera eficiente.

Una de las principales razones para su elección en este proyecto es la disponibilidad de una amplia gama de paquetes, muchos de ellos diseñados para necesidades muy específicas. En particular, el paquete BiblioMetrics, cuyas funcionalidades se implementarán en el desarrollo del proyecto, no cuenta con un equivalente en otros paquetes disponibles en el repositorio **CRAN** [4], lo que resalta su relevancia para los objetivos planteados.

Además, durante la implementación, fue necesario utilizar otros lenguajes para desarrollar la interfaz y realizar *web scrapping*, ya que no había APIs disponibles para consultar los datos.

[2] [Web de Trello]. (s.f.). Trello. https://trello.com/. Fecha de acceso: 25 de abril de 2024.
[3] [Manual de R]. (s.f.). CRAN. https://cran.r-project.org/. Fecha de acceso: 25 de abril de 2024.
[4] [Listado del repositorio de paquetes R existentes ordenados por fecha]. (s.f.). CRAN. https://cran.r-project.org/. Fecha de acceso: 25 de abril de 2024.

### CSS (Cascading Style Sheets)

En español, una **hoja de estilos** es el lenguaje utilizado para controlar la apariencia del diseño de una página web. Con él, se puede modificar el estilo de la interfaz gráfica para darle una apariencia profesional y original.

### Javascript

El lenguaje **JavaScript** está diseñado para editar el comportamiento de la interfaz, crear interactividad con la aplicación y ofrecer una experiencia de usuario dinámica.

En el caso de BiblioMetrics, se ha utilizado para mostrar y ocultar partes de la interfaz según ciertas condiciones y para crear los *sliders* numéricos que aparecerán cuando se realiza una búsqueda con datos numéricos.

### HTML en R

**HTML** es un lenguaje de marcas, llamado así porque utiliza etiquetas ”<>” para estructurar el contenido en el navegador web. Este lenguaje se usa para definir la estructura y el contenido visual de una página web o interfaz gráfica.

En el desarrollo de BiblioMetrics, realizado en R, como se mencionó anteriormente, la interfaz gráfica (`UI`) se ha construido de manera convencional utilizando el paquete Shiny 7.2.2. Gran parte de esta interfaz ha sido diseñada con *tags* HTML, las cuales han organizado y definido la apariencia visual para el usuario.

### Python

**Python** es un lenguaje de programación muy utilizado en el trabajo con datos y en *machine learning*. En este caso, se ha empleado exclusivamente para realizar **web scrapping** y obtener datos de algunas bases de datos bibliométricas que no contaban con API o no permitían acceder a esos datos de otra manera. Aunque la idea inicial no era utilizar este lenguaje, los problemas encontrados obligaron a recurrir a él.

### 7.1.4. Entorno de Desarrollo Integrado (IDE)

Nuestro IDE[5] será **RStudio**, este tiene una interfaz de usuario intuitiva, la cual facilita la escritura, ejecución y depuración de código. Además, también incluye funciones integradas, lo que facilita el trabajo. Se integra bastante bien también con herramientas para el control de versiones. Incluye también herramientas para la generación de gráficos y es bastante extensible, gracias a complementos.

La versión *Desktop* que es la que se ha trabajado en este proyecto, está disponible para Windows, Mac y Linux. En este caso, se trabajará para el sistema operativo Windows 11 Home (10.0.22631).

### 7.1.5. Herramienta para control de versiones

GitHub[6] es una plataforma en línea que utiliza Git y permite la colaboración en proyectos de software mediante el sistema de control de versiones. Es ampliamente utilizado tanto en el ámbito empresarial como en el científico.

Una de las características más importantes de Git es la capacidad de revertir cambios en el código en caso de fallos o problemas inesperados durante el desarrollo del proyecto. Esto proporciona seguridad y permite corregir errores de manera rápida y eficiente. Otra característica significativa es su capacidad para facilitar la colaboración, permitiendo que el cliente-tutor esté al tanto de todo el proceso de implementación del proyecto. Se elige GitHub debido a su integración con el IDE (7.1.4) y porque el desarrollador está acostumbrado a trabajar con esta plataforma.

## 7.2. Paquetes de R utilizados

Para el desarrollo del paquete, ha sido necesario utilizar una serie de paquetes para la ayuda tanto de la interfaz gráfica, como de los inconvenientes que se han ido obteniendo mientras se llevaba a cabo el desarrollo.

[5] Software que ayuda a escribir, editar y ejecutar código de manera eficiente
[6] [Web de GitHub]. (s.f.). GitHub. https://github.com/. Fecha de acceso: 25 de abril de 2024.

### 7.2.1. Shiny

**Shiny** [7] es un paquete de R que permite crear aplicaciones web interactivas directamente desde R. Facilita tanto la construcción de interfaces de usuario (`UI`) como la implementación de la funcionalidad del servidor (`Server`) para desarrollar aplicaciones dinámicas. Por lo tanto, utilizaremos este paquete para crear nuestra interfaz gráfica, ya que además de facilitar este proceso, ofrece una excelente integración con otros paquetes, lo que permite diseñar una interfaz muy completa.

### 7.2.2. Shinyjs

**Shinyjs** [8] es un paquete totalmente complementario a **Shiny** que permite añadir funcionalidades **JavaScript** a las aplicaciones, mejorando la interactividad y controlando comportamientos dinámicos. En el contexto de BiblioMetrics, la función de este paquete es mejorar la experiencia del usuario, proporcionando retroalimentación inmediata.

### 7.2.3. Shinymaterial

**Shinymaterial** [9] es un paquete no oficial que integra algunas de las propiedades básicas del framework de diseño **Materialize CSS** [10] en aplicaciones Shiny. Para el desarrollo de la interfaz, se ha utilizado como complemento, ya que facilita la creación de ciertos elementos de Materialize. Sin embargo, también se han empleado las etiquetas `tags` de HTML en R y el atributo `global class` de HTML para añadir las clases específicas utilizadas por Materialize. Esto ha permitido combinar la flexibilidad de Shiny con el diseño estilizado de Materialize CSS, aportando a la interfaz un diseño minimalista, robusto y de apariencia profesional.

[7] [Shiny] CRAN. https://shiny.posit.co/
[8] Shinyjs CRAN. https://cran.r-project.org/web/packages/shinyjs/index.html
[9] CRAN shinymaterial https://cran.r-project.org/web/packages/shinymaterial/index.html
[10] Materialize. https://materializecss.com/

### 7.2.4. DT

**DT** [11] es un paquete de R que proporciona una interfaz para utilizar la librería **JavaScript DataTables** [12] en aplicaciones Shiny. Esto permite crear tablas interactivas y personalizables, facilitando el desarrollo, la visualización y la manipulación de datos de manera eficiente.

### 7.2.5. Plotly

**Plotly** [13] es un paquete de R que permite crear gráficos interactivos y dinámicos. Basado en la biblioteca **JavaScript Plotly.js** [14], este paquete soporta una gran diversidad de tipos de gráficos. Además, aporta interactividad a los gráficos, ofreciendo al usuario una experiencia más agradable y facilitando la exploración y el análisis de datos de manera intuitiva y atractiva.

### 7.2.6. Stringr

El paquete **Stringr** [15], es usado para manipular y procesar cadenas de texto en R. En este proyecto, es usado para limpiar texto de entrada, eliminando espacios y caracteres innecesarios.

### 7.2.7. Writexl

**Writexl** [16] es un paquete que permite escribir datos en archivos Excel. Para este proyecto, se utiliza para exportar los datos solicitados por el usuario a un archivo Excel para ser descargado

[11] CRAN:Package DT https://cran.r-project.org/web/packages/DT/index.html
[12] JavaScript DataTables https://datatables.net/
[13] CRAN:Package Plotly https://cran.r-project.org/web/packages/plotly/index.html
[14] Plotly.js https://plotly.com/
[15] CRAN:Package stringr https://cran.r-project.org/web/packages/stringr/index.html
[16] CRAN:Package writexl https://cran.r-project.org/web/packages/writexl/index.html

### 7.2.8. Httr

El paquete **Httr** [17] se encarga del manejo de solicitudes HTTP desde R. Es útil para trabajar con datos de APIs web, ya que permite realizar métodos HTTP como *GET*, *POST*, *PUT* y *DELETE*. En el proyecto BiblioMetrics, se ha utilizado httr para recoger información de diversas APIs web, facilitando la obtención de datos necesarios.

### 7.2.9. Jsonlite

La función **Jsonlite** [18], se utiliza para convertir datos en formato JSON a estructuras de datos de R, facilitando así su manipulación y uso. Esta funcionalidad es particularmente valiosa cuando se obtienen datos de APIs web, que suelen exportar información en formato JSON. Además, jsonlite es empleada en la mayoría de los métodos del paquete para manejar archivos de configuración y otros aspectos relacionados.

### 7.2.10. Reticulate

**Reticulate** [19], es un paquete, el cual permite añadir funcionalidades del lenguaje Python 7.1.3, ejecutar código, importar módulos y compartir datos entre R y Python. El uso en este proyecto, es obligado por los problemas de acceso a datos de APIs que serán descritos más adelante, obligando a usar *web scrapping*, para obtener datos, mediante la biblioteca de Python **"Selenium"** y sus diversos métodos implementados.

## 7.3. Fuentes de datos bibliométricos

Para obtener los datos bibliométricos necesarios para el análisis, como se menciona en la introducción 1.1, se utilizan tres bases de datos principales: **Scopus**, **Web of Science** y **Google Scholar**. Se planea acceder a estas fuentes para recolectar información sobre artículos, revistas y autores, así como para obtener datos bibliométricos específicos de cada uno.

[17] CRAN:Package httr https://cran.r-project.org/web/packages/httr/index.html
[18] CRAN:Package jsonlite https://cran.r-project.org/web/packages/jsonlite/index.html
[19] CRAN:Package reticulate https://cran.r-project.org/web/packages/reticulate/index.html

Durante el proceso de desarrollo, se presentaron dificultades significativas para acceder a la información de varias bases de datos, las cuales se detallarán en los apartados correspondientes. Estos problemas no solo afectaron la planificación del proyecto, añadiendo una carga de trabajo adicional, sino que también impidieron que la aplicación ofreciera la funcionalidad completa para todos los tipos de búsqueda inicialmente previstos. Como resultado, la aplicación no cumple en su totalidad con las expectativas de funcionalidad, lo que la deja incompleta en comparación con la idea inicial.

### 7.3.1. Scopus

Ha sido la base de datos más versátil, ya que la cuenta institucional de la Universidad de Jaén me ha permitido acceder a numerosas APIs. Sin embargo, al intentar obtener métricas de un artículo específico, la API diseñada para ese propósito era de pago. Como resultado, se implementó un método de *web scrapping* para acceder a los datos necesarios.

### 7.3.2. Web Of Science

En el intento de acceder a los datos de la base de datos bibliométrica Web of Science (WOS), disponible a través de una cuenta institucional proporcionada por la universidad, se han enfrentado diversas limitaciones durante el proceso de obtención de los datos necesarios para el análisis. Estas limitaciones han afectado el avance del proyecto.

En primer lugar, la cuenta institucional ofrecida solo permite el acceso a una API de obtención de datos muy limitada. Esta API restringe severamente la cantidad de artículos que se pueden recuperar y no proporciona métricas necesarias para llevar a cabo el análisis previsto en el proyecto.

Además, la plataforma no ofrece acceso gratuito a APIs que permitan la recuperación de datos de autores o de revistas. Estas restricciones han limitado considerablemente el alcance y la profundidad del análisis bibliométrico que se pretende realizar.

Para superar estas limitaciones, se realizó una consulta con la universidad para explorar la posibilidad de obtener un acceso más amplio a través de sus canales oficiales. No obstante, la respuesta obtenida fue que no era posible ampliar el acceso a

las funcionalidades requeridas para el proyecto, lo que dejó a la investigación en una situación sin alternativas viables desde el acceso institucional.

Como última alternativa, se intentó emplear técnicas de *web scrapping* para extraer datos directamente de la plataforma de WOS. Sin embargo, este enfoque también resultó errado. La principal dificultad radica en que, para acceder a ciertos datos detallados, es necesario estar conectado desde una red corporativa universitaria. Fuera de esta red, muchos de los datos necesarios no están disponibles. Además, la estructura y el diseño de la página web de WOS hacen que el proceso de *web scrapping* sea extremadamente complejo y desafiante, limitando aún más la viabilidad de esta metodología.

Por tanto, desde BiblioMetrics no será posible realizar el análisis previsto, quedando esta funcionalidad como una posible área de mejora para futuras actualizaciones. Aunque la opción de elegir la base de datos bibliométrica estará disponible para los usuarios, la funcionalidad asociada a ella permanecerá bloqueada y, por ende, inaccesible en la versión actual.

### 7.3.3. Google Scholar

En cuanto a la base de datos bibliométrica Google Scholar, se observó desde el inicio que no era una de las opciones más destacadas y, además, carecía de una API oficial para la obtención de datos. A pesar de ello, se realizó una revisión exhaustiva de la plataforma y se constató que, aunque no proporcionaba métricas relacionadas con artículos y revistas, sí ofrecía información métrica relevante sobre autores.

Dado que Google Scholar no proporcionaba métricas de artículos y revistas que pudieran ser analizadas en el proyecto, estas se descartaron como fuentes de datos. No obstante, se identificó que las métricas de los autores estaban disponibles y, por lo tanto, se procedió a implementar métodos de *web scraping* para extraer esta información. En las secciones siguientes se detallarán los métodos utilizados para obtener datos de autores a través de consultas y, posteriormente, para recopilar métricas específicas sobre cada autor.

## 7.4. Desarrollo de interfaz (UI)

Una vez explicado detenidamente el contexto y los elementos con los que se ha trabajado en este proyecto, se procederá a detallar cada uno de los desarrollos realizados. En primer lugar, se abordará el desarrollo de la interfaz de usuario (`UI`), que incluye el diseño y maquetación del sitio, así como la creación de elementos interactivos que, mediante JavaScript, proporcionan una experiencia dinámica al usuario.

Es relevante destacar que, como se mencionó anteriormente, se ha utilizado el *framework* Materialize 7.2.3 como base para este desarrollo. Este framework fue analizado a fondo y se consultó su guía para su adecuado aprendizaje y aplicación. La biblioteca **Shinymaterial** fue fundamental en este proceso, ya que, aunque no proporcionaba todas las funcionalidades necesarias para evitar el uso de `tags` HTML en R, resultó ser una herramienta de gran utilidad. La elección de este *framework* se basó en su capacidad para ofrecer una interfaz profesional y estéticamente agradable, facilitando una experiencia de usuario intuitiva y visualmente atractiva.

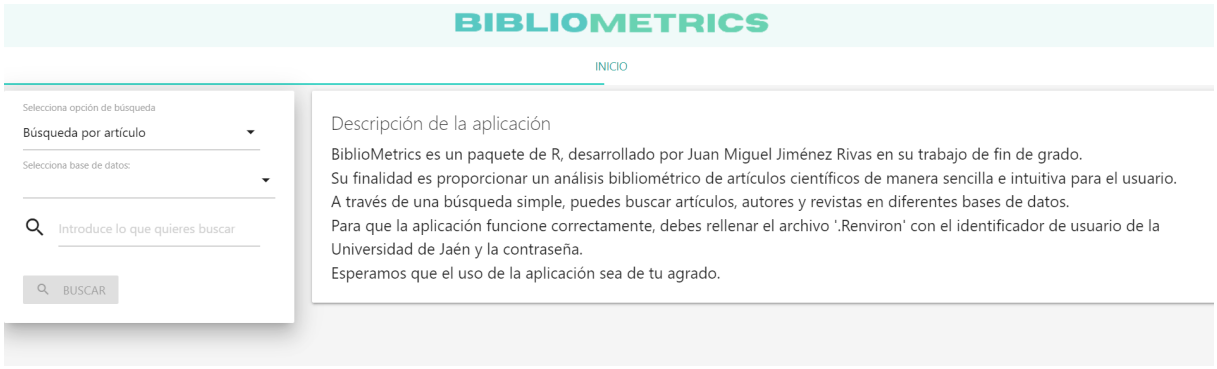



Figura 7.1: Pantalla de inicio de interfaz gráfica BiblioMetrics
**Fuente: Elaboración propia.**

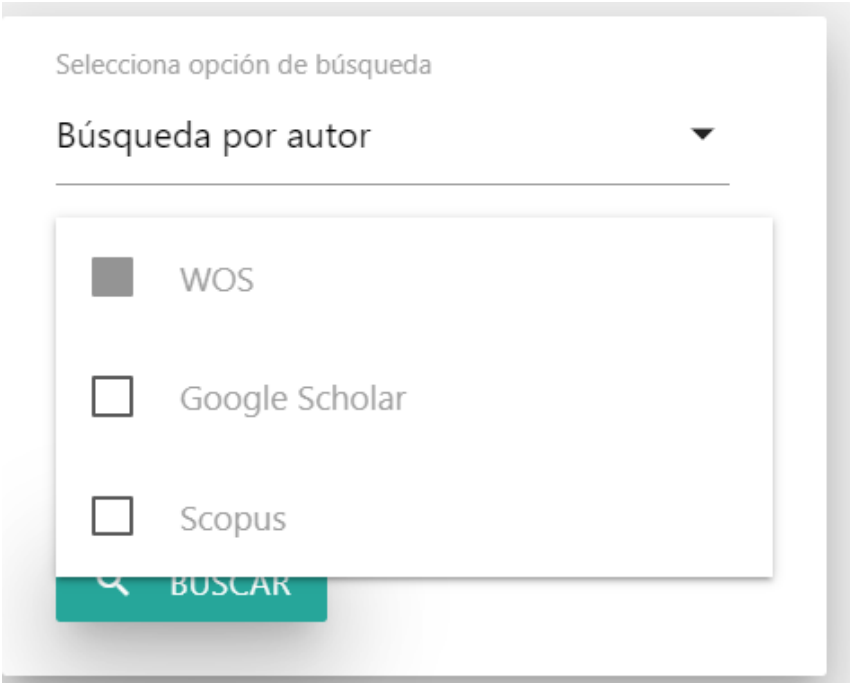



Figura 7.2: Captura filtro elección bases de datos de interfaz gráfica
**Fuente: Elaboración propia.**


Figura 7.3: Captura filtro elección de tipo de búsqueda en interfaz gráfica
**Fuente: Elaboración propia.**

### 7.4.1. Inconvenientes surgidos

En resumen, en relación con esta sección del proyecto (aunque en R no se utilizan clases en el sentido tradicional de la programación orientada a objetos), los problemas enfrentados han sido principalmente de naturaleza HTML y CSS. Cada error en estos lenguajes ha requerido un análisis detallado para identificar y comprender su causa y posible solución. Como se puede observar en las imágenes anteriores, se ha logrado replicar de manera aceptable el diseño del *mock-up* 6.4 previamente elaborado, cumpliendo con las expectativas establecidas en términos de presentación y funcionalidad.

## 7.5. Desarrollo de Server

El desarrollo del componente `Server` ha presentado un nivel de complejidad considerable. Este componente es fundamental para establecer la interacción entre la interfaz de usuario (`UI`) y el conjunto de métodos del paquete, los cuales se encargan de procesar las consultas y devolver los datos solicitados. Además, el servidor gestiona la funcionalidad dinámica del sistema, controlando la visibilidad y el estado interactivo de los elementos de la interfaz. En particular, el servidor se ocupa de habilitar y deshabilitar elementos, así como de mostrar u ocultar componentes según las acciones del usuario y el estado de las consultas. Este enfoque permite crear una experiencia de usuario fluida y coherente, minimizando el margen de error y garantizando una interacción intuitiva.

### 7.5.1. Observador del botón de búsqueda

El componente `Server` es responsable de supervisar la activación del botón de búsqueda. Cuando el botón es presionado, el componente captura los elementos seleccionados por el usuario, incluyendo el tipo de consulta que se desea realizar (ya sea artículo, autor o revista) y las bases de datos en las que se desea realizar la búsqueda. Con esta información, el componente `Server` llama al método correspondiente del paquete BiblioMetrics para procesar la consulta y obtener los datos necesarios.

Además, dentro de este componente, se generan filtros específicos para cada tipo de consulta, adaptados a los datos que se esperan recibir. Finalmente, el componente

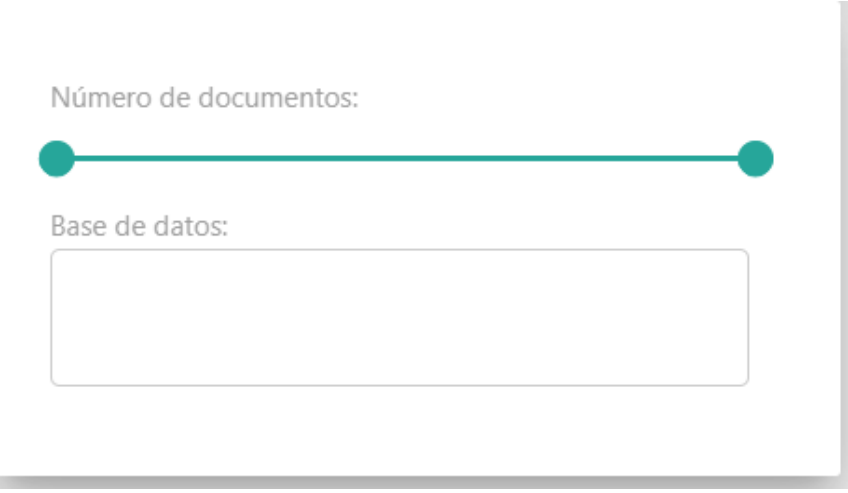



Figura 7.4: Captura de filtros de una consulta específica en interfaz gráfica
**Fuente: Elaboración propia.**

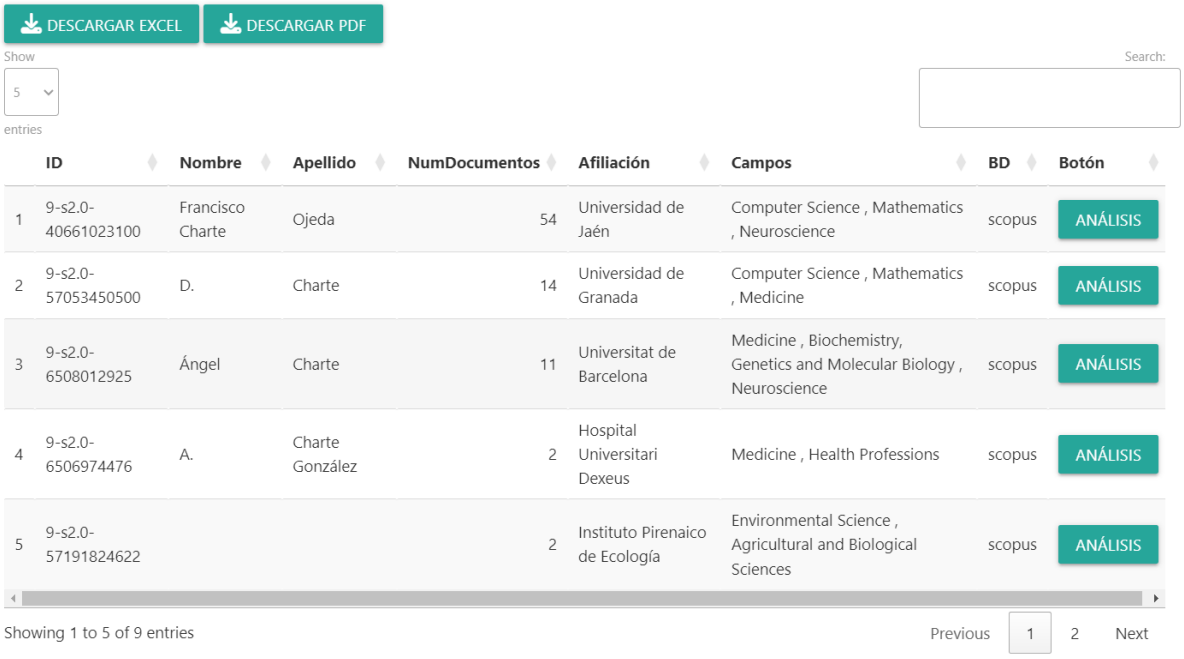
`Server` crea una tabla interactiva que presenta los datos obtenidos de manera clara y accesible para el usuario.

### 7.5.2. Filtrado de datos

Este componente también incluye una función reactiva, que se encarga de aplicar los filtros elige el usuario, que han sido generados por cada consulta en el observador del botón de búsqueda. Esta función reactiva ajusta el conjunto de datos que se mostrará en la tabla, asegurando que solo se presenten los resultados filtrados según los criterios seleccionados por el usuario.

### 7.5.3. Renderizado de tabla de resultados

El componente `Server` también es responsable de mantener un observador que detecta cualquier cambio en los filtros. Cuando se produce un cambio, la tabla se actualiza de forma reactiva para mostrar los datos filtrados correspondientes.



DESCARGAR EXCEL DESCARGAR PDF

Show 5 entries

Search:

| | ID | Nombre | Apellido | NumDocumentos | Afiliación | Campos | BD | Botón |
|---|---|---|---|---|---|---|---|---|
| 1 | 9-s2.0-40661023100 | Francisco Charte | Ojeda | 54 | Universidad de Jaén | Computer Science , Mathematics , Neuroscience | scopus | ANÁLISIS |
| 2 | 9-s2.0-57053450500 | D. | Charte | 14 | Universidad de Granada | Computer Science , Mathematics , Medicine | scopus | ANÁLISIS |
| 3 | 9-s2.0-6508012925 | Ángel | Charte | 11 | Universitat de Barcelona | Medicine , Biochemistry, Genetics and Molecular Biology , Neuroscience | scopus | ANÁLISIS |
| 4 | 9-s2.0-6506974476 | A. | Charte González | 2 | Hospital Universitari Dexeus | Medicine , Health Professions | scopus | ANÁLISIS |
| 5 | 9-s2.0-57191824622 | | | 2 | Instituto Pirenaico de Ecología | Environmental Science , Agricultural and Biological Sciences | scopus | ANÁLISIS |

Showing 1 to 5 of 9 entries Previous 1 2 Next

Figura 7.5: Captura de tabla de datos de una consulta específica en interfaz gráfica
**Fuente: Elaboración propia.**

### 7.5.4. Observadores de botones de descargas

Este componente también incluye una función que observa si alguno de los dos botones de descarga de datos ha sido pulsado. Estas funciones se encargan de exportar los datos que se están mostrando en ese momento y permiten al usuario descargarlos en formato PDF o XLSX.

### 7.5.5. Observador de botón de análisis de la tabla

En este componente, al crear la tabla, se genera un botón para cada fila de datos, como se puede observar en la figura 7.5. Cuando se pulsa uno de estos botones, se desencadena un evento, que es detectado por esta parte del código. Este código se encarga de enviar la consulta previamente realizada y la fila de datos seleccionada al método correspondiente del paquete BiblioMetrics. A continuación, se recuperan los datos métricos, y esta parte del código realiza un análisis, mostrando gráficos y datos visuales, si es posible, para facilitar su interpretación.

### 7.5.6. Inconvenientes surgidos

Debido a la extensión y complejidad de este método, han surgido una serie de errores y problemas que han causado retrasos en la planificación. En particular, la actualización de los filtros y los datos de la tabla ha requerido una cantidad considerable de tiempo durante la implementación.

## 7.6. Desarrollo de métodos BiblioMetric

Los métodos de BiblioMetrics son responsables de procesar las consultas obtenidas por la ”clase” `Server` 7.5, realizando las consultas correspondientes a la base de datos bibliométrica de manera específica según el tipo de consulta. De este modo, el componente `Server` puede encargarse tanto de devolver la tabla de datos, si es necesario, como de generar la ventana de análisis con los datos obtenidos de la base de datos.

### 7.6.1. getArticles

Este método es responsable de devolver los datos de artículos obtenidos inicialmente desde cualquier base de datos bibliométrica. Sin embargo, debido a las limitaciones en los diferentes métodos de obtención de datos, en esta sección solo se mostrará la integración con Scopus activa, y únicamente se podrán obtener datos de esa base de datos.

#### Scopus

El método `getArticlesScopus()` consultará la variable de texto proporcionada por el usuario, buscando documentos en los que dicha variable aparezca en el título, palabras clave o resumen. Se obtendrán los 50 resultados más relevantes, y si hay resultados, se creará un *dataframe* con los siguientes datos:

- ID
- Título
- Autor
- Año de publicación
- DOI
- Número de citas totales
- Base de datos de donde ha sido obtenida

Si no se encuentran resultados, el *dataframe* devuelto por defecto en todos los métodos tendrá una sola columna llamada `error`, con un valor lógico igual a `TRUE`.

### 7.6.2. getAuthors

El método `getAuthors()` se ejecutará de acuerdo con la API o APIs proporcionadas como parámetro y llamará a los métodos necesarios para obtener los datos de los autores de la base de datos correspondiente (ya sea Google Scholar o Scopus). Finalmente, devolverá un *dataframe* con columnas comunes, que se ajustarán al formato de la base de datos bibliométrica utilizada.

| Tipo dato devuelto | Scopus | Google Scholar |
|---|---|---|
| ID | X | X |
| Nombre | X | X |
| Apellidos | | X |
| Afiliación | X | X |
| Citas | X | |
| Campos | | X |
| Base de datos | X | X |

Tabla. 7.1: Datos dataframe de autores

#### Google Scholar

Como se explicó en el apartado 7.3.3, no fue posible obtener los datos a través de una API oficial, ya que no existe una API oficial y las disponibles eran muy limitadas y no proporcionaban datos métricos. Por lo tanto, se optó por implementar el método en Python, utilizando el paquete `reticulate` y la librería `selenium`. Este método consiste en realizar una consulta en un enlace específico, introducir la consulta del usuario y obtener los elementos HTML que corresponden a los autores. A partir de estos elementos, se extraen todos los datos presentados en la tabla anterior 7.1.

#### Scopus

En este caso, al consultar la información de la API de la base de datos Scopus, se devuelven los autores cuyos apellidos coinciden con el texto proporcionado por el usuario. La API devuelve un máximo de 50 resultados, y el *dataframe* resultante tendrá el formato descrito en la tabla 7.1.

### 7.6.3. getSources

Al igual que en el método de recopilación de artículos, ha habido problemas para obtener datos de revistas, lo que ha resultado en que solo se pueda consultar la base de datos de Scopus. Este método, por lo tanto, toma la consulta del usuario y devuelve un *dataframe* con todas las revistas y sus datos que coinciden con la consulta.

#### Scopus

En este método, `getSourcesScopus()`, se realiza una consulta a la API de la base de datos bibliométrica Scopus para obtener información sobre las revistas que coinciden con la consulta introducida por el usuario. Como resultado, se devuelve un *dataframe* con los siguientes datos:

- ID
- Título
- Año de publicación
- Campos
- Base de datos de donde ha sido obtenida

### 7.6.4. getMetricsArticle

Dado que el método `getArticle()` en la sección 7.6.1 solo utiliza la base de datos de Scopus, el apartado que obtiene los datos de uno de los artículos devueltos por el *dataframe* de este método también estará limitado a la base de datos bibliométrica Scopus.

#### Scopus

En este caso, Scopus no ofrece APIs con métricas de artículos a menos que se adquiera una versión de pago, ya que dichas métricas son proporcionadas por SciVal[20]. Por lo tanto, la única forma de obtener estas métricas, para cumplir con el requisito de contar al menos con un método para obtener métricas de artículos, era mediante *web scrapping*. Sin embargo, para llevar a cabo este *web scrapping*, se requieren credenciales corporativas de la Universidad de Jaén, ya que el acceso a los datos de Scopus a través de la web requiere iniciar sesión. Dado que cada inicio de sesión en sitios corporativos es único, el método de *web scrapping* está diseñado específicamente para el acceso de la Universidad de Jaén. El método navega hasta el enlace del artículo utilizando su identificador y, a partir de allí, extrae todas las métricas disponibles.

[20]SciVal: es una herramienta de pago de análisis e investigación que se basa en datos de Scopus https://www.elsevier.com/es-es/products/scival

El método de Python, devuelve un diccionario, el cual está orientado de la siguiente manera:

- Métricas de Scopus:
  - citaciones en Scopus
  - percentil
  - FWCI: es un orientador normalizado de Scopus
- Contador de visitas:
  - visitas totales
  - años
- Metricas plumx:
  - capturas
    - lectores
  - menciones
    - nuevas menciones
    - menciones en blog
    - referencias
  - social
    - veces que lo han compartido, le han dado like o han comentado

### 7.6.5. getMetricsAuthor

El método `getMetricsAuthor()` seleccionará la base de datos de la cual recogerá la métrica, en función de la opción elegida por el usuario, y devolverá un *dataframe* diferente según la base de datos consultada. Sin embargo, siempre garantiza la devolución de un *dataframe*.

#### Scholar

Para obtener los datos métricos de Google Scholar, si antes he usado *web scrapping* para obtener los datos, ahora no iba a ser menos, lo que se hará es introducir dentro de la página de cada autor en Google Scholar y se recogerán los elementos

con datos interesantes para realizar análisis. En este caso, el método Python devuelve una lista, con diferentes diccionarios, los datos comunes a obtener serán en forma de tabla tal que:

| **Métrica** | **Total** | **Desde 2019** |
|---|---|---|
| Citas | X | X |
| Índice h | X | X |
| Índice 10 | X | X |

Tabla. 7.2: Datos devueltos por getMetricsAuthorScholar()

#### Scopus

Para acceder a los datos en Scopus, utilizaremos la API de contenido de autores, empleando el identificador de Scopus que se obtuvo durante la consulta de autores. A partir de esto, se generará un *dataframe* con los siguientes campos:

- ORCID
- Nombre
- Número de documentos
- Número de citas
- Número de documentos citados
- Índice H
- Número de veces que ha sido co-autor
- Universidad
- Lugar de la universidad

### 7.6.6. getMetricsSource

Este método, al igual que en el caso de obtener datos de revistas, solo permite consultar la base de datos bibliométrica Scopus. Por lo tanto, solo se podrán obtener métricas de esta base de datos. El resultado será una lista de *dataframes* con numerosos datos para analizar.

### Scopus

Este método es uno de los más complejos para obtener información, ya que pueden presentarse dos situaciones:

1. Si la revista a buscar tiene un ISSN, la búsqueda se realizará directamente por este identificador.
2. Si la revista no tiene ISSN, se buscará información utilizando el nombre de la revista.

En ambos casos, se utilizará la API de Scopus denominada *"Serial Title API"*. Lo que será común en la búsqueda es la información que obtendremos de esta API, que se presentará en una lista de *dataframes* y otras estructuras de datos, ya que los datos son diferentes e independientes entre sí.

El primer *dataframe* incluirá los siguientes campos:

- ISSN
- Editorial
- Título
- Año de inicio de la revista

A continuación, se proporcionará una lista de todas las áreas temáticas asociadas con la revista.

Después, habrá otro *dataframe* que devolverá las métricas de impacto de Scopus para las revistas, incluyendo SNIP [21] y SJR [22], junto con el año en que se obtuvo cada métrica.

Finalmente, se generará un *dataframe* con datos anuales sobre la revista, que mostrará para cada año:

- Número de documentos

[21] SNIP (Source Normalized Impact per Paper): se puede definir como el número de citas medio recibido por los artículos de una revista durante tres años (Raw impact per paper RIP) dividido entre la citación potencial del campo científico de la revista (Relative database citation potential RDCP).[10]

[22] SJR: cálculo de las citas recibidas por las revistas en un periodo de 3 años, otorgando un peso mayor a las citas procedentes de revistas de alto prestigio (aquellas con altas tasas de citación y baja autocitación) utilizando para ello el algoritmo de Google PageRank. [11]

- Número de citas
- Número de documentos no citados
- Porcentaje de documentos no citados
- Porcentaje de artículos en revisión

# Capítulo 8

# APÉNDICE: MANUALES DE USO

## 8.1. Apendice A. Manual de instalación

Este manual, mostrará la información para llevar a cabo la instalación del paquete partiendo desde cero.

### 8.1.1. Descargar versión de R

Para utilizar BiblioMetrics, es fundamental contar con una versión actualizada de R. Si aún no tienes R instalado en tu sistema, el primer paso es acceder al sitio web oficial en https://www.r-project.org/. Una vez allí, haz clic en el enlace ”download” 8.1.

A continuación, serás redirigido a https://cran.r-project.org/mirrors.html, donde deberás seleccionar un servidor que corresponda a tu ubicación geográfica. En este manual, se emplearán servidores ubicados en España. Al acceder a uno de estos servidores, verás una página similar a la imagen 8.2, donde deberás elegir el enlace correspondiente a tu sistema operativo.

En este punto, se desplegarán diferentes subdirectorios. Deberás seleccionar el indicado en la figura 8.3 para descargar el código base de R..

Una vez completados los pasos anteriores, aparecerá una nueva página donde deberás pulsar en el enlace que pone **”Download R (la versión a descargar) for (el sistema operativo donde se quiere descargar)** y tras esto comenzará la descarga de R.

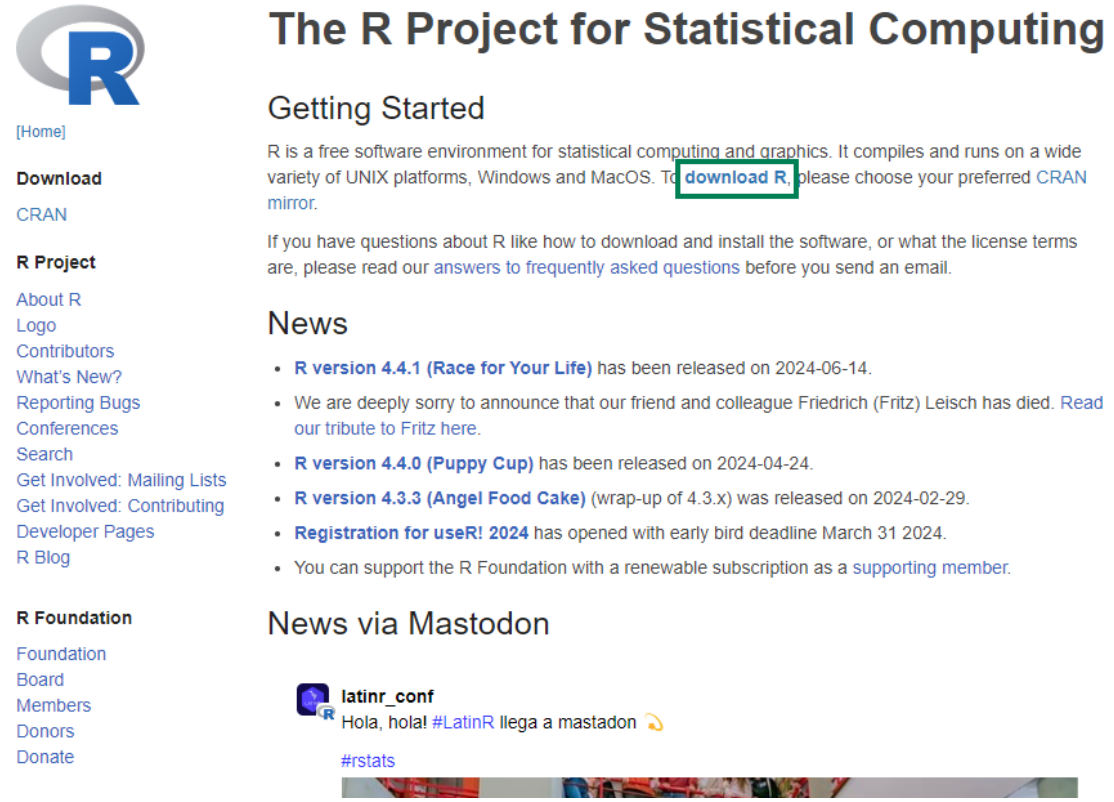



Figura 8.1: Página inicial R-project
**Fuente: Elaboración propia.**


Figura 8.2: Página de elección de nuestro sistema operativo
**Fuente: Elaboración propia.**

Una vez descargado el archivo, procede a ejecutarlo.

A partir de este punto, el proceso de instalación puede variar según el sistema operativo. En este caso, como se ha mencionado y mostrado previamente, el manual

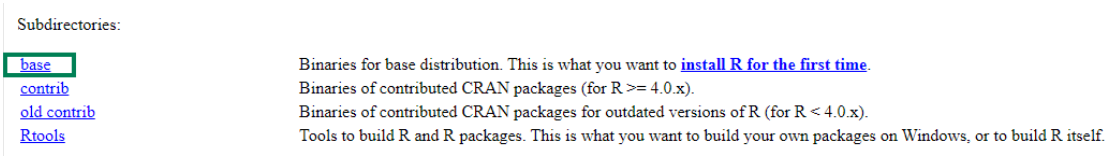



Figura 8.3: Captura para seleccionar código base de R
**Fuente: Elaboración propia.**

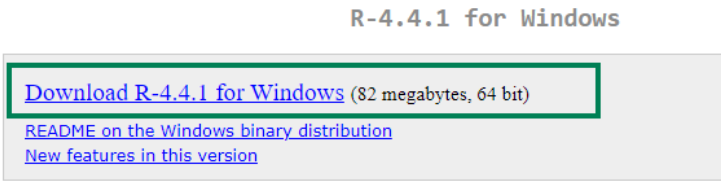



Figura 8.4: Selección de enlace de descarga
**Fuente: Elaboración propia.**

se lleva a cabo en un entorno Windows, específicamente en la versión Windows 11 Pro.

Al ejecutar el archivo, lo primero que aparecerá será una consulta del sistema operativo, solicitando tu confirmación para permitir que la aplicación realice cambios en el equipo. Para continuar con la instalación de R, deberás aceptar esta solicitud. Es importante destacar que necesitas una cuenta con permisos de administrador; si no dispones de una, será necesario que un usuario con privilegios de administrador intervenga en esta etapa del proceso.

En la siguiente ventana, se te dará la opción de elegir el idioma de instalación. Para este proyecto, seleccionaremos Español y haremos clic en ”Aceptar”.

A continuación, aparecerá una ventana con los términos y condiciones, los cuales deberás aceptar (Figura 8.5) para poder continuar con la instalación.

Después de este paso, el instalador te solicitará que elijas la ubicación donde se instalará R. Por defecto, se propondrá una ruta (la cual se utilizará en este manual), pero el usuario también tiene la opción de seleccionar una ruta alternativa, para confirmar se pulsará el botón ”Siguiente” (Figura 8.6).

Después de este paso, el instalador te pedirá que selecciones los componentes que deseas instalar, permitiéndote marcar o desmarcar las opciones disponibles. En este manual, instalaremos todos los componentes que se muestran en la figura 8.7, y luego haremos clic en ”Siguiente”.

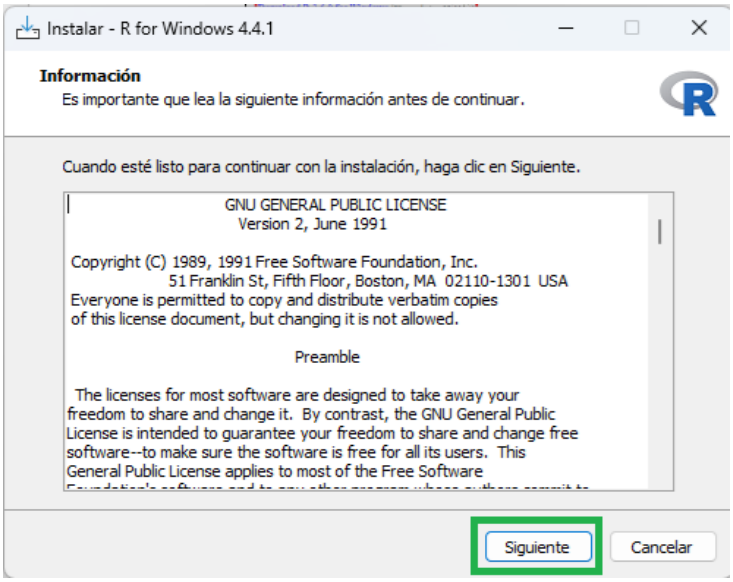



Figura 8.5: Aceptación de términos y condiciones de R
**Fuente: Elaboración propia.**

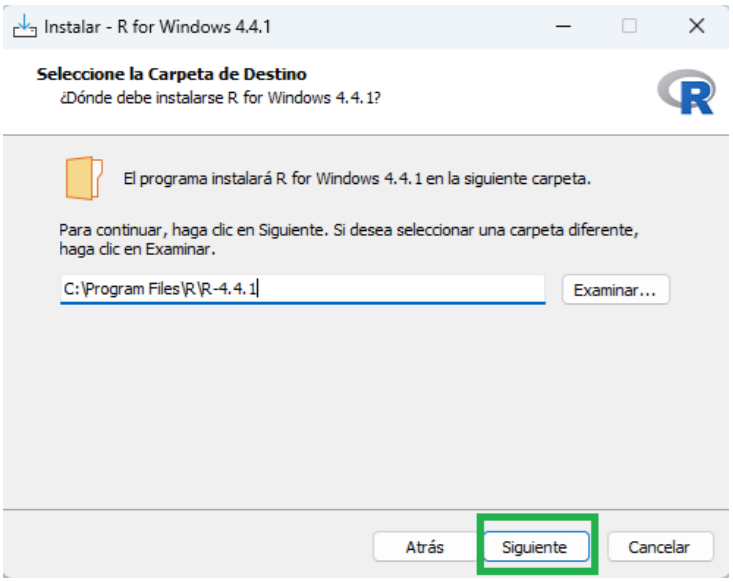



Figura 8.6: Captura de elección de ruta de instalación de R
**Fuente: Elaboración propia.**

La siguiente ventana, te preguntará si deseas utilizar opciones de configuración personalizadas. Si seleccionas ”Si”, podrás ajustar la configuración de R según tus preferencias. Sin embargo, en este manual utilizaremos la configuración predeterminada, por lo que seleccionaremos ”No” y procederemos con la instalación.

Posteriormente, el instalador te preguntará dónde deseas que se creen los accesos directos del programa en la carpeta del Menú Inicio. Si lo dejas en la opción predeterminada, se crearán en una carpeta llamada ”R”. También tienes la opción de no crear una carpeta en el Menú Inicio o de cambiar el nombre de la carpeta. En este

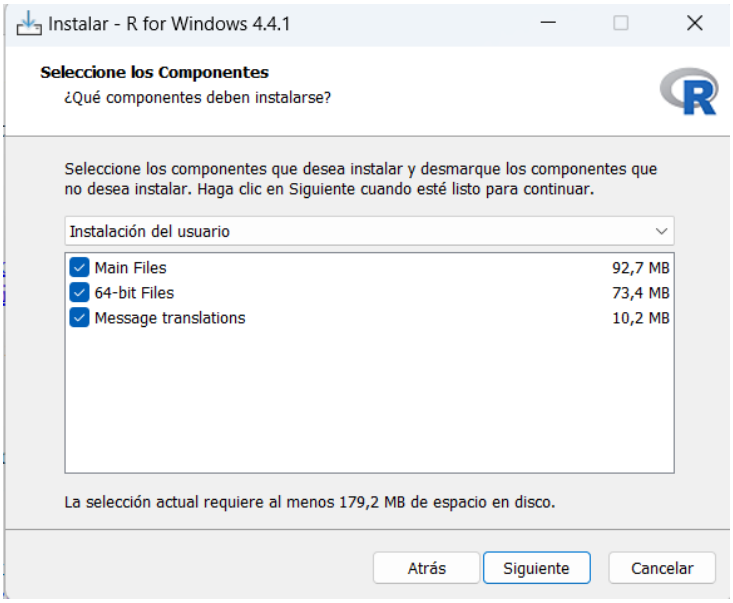



Figura 8.7: Captura de elección de componentes en la instalación de R
**Fuente: Elaboración propia.**

manual, dejaremos la configuración por defecto, creando la carpeta ”R” en el Menú Inicio (Figura 8.8).

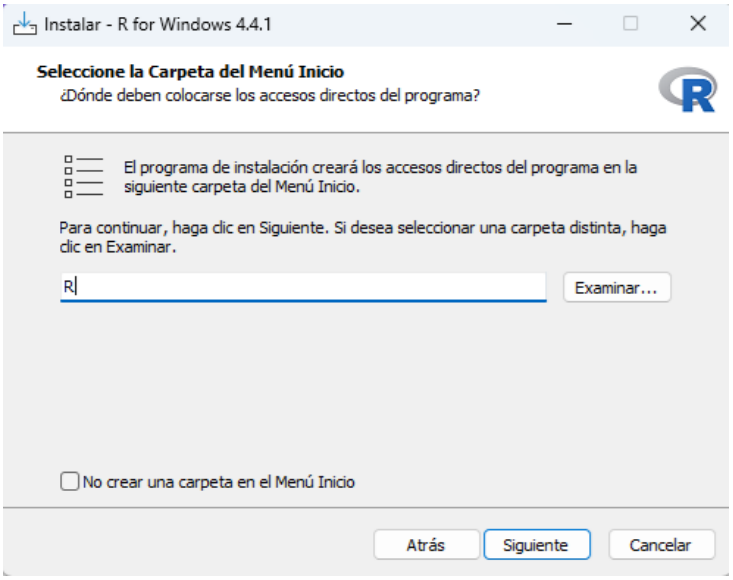



Figura 8.8: Captura de elección de nombre menú inicio
**Fuente: Elaboración propia.**

Finalmente, antes de comenzar la instalación de R en tu equipo, el instalador te permitirá seleccionar tareas adicionales que deseas realizar durante el proceso. Estas tareas incluyen la creación de accesos directos adicionales (en el escritorio y en Inicio Rápido), el guardado de un registro con el número de versión, y la asociación de archivos con la extensión ”.RData” a R. Estas opciones pueden ajustarse según las preferencias del usuario. En este manual, crearemos un acceso directo en el escrito-

rio, guardaremos el número de versión en el registro, y configuraremos para que los archivos con la extensión ".RData" se asocien con R.


Figura 8.9: Captura de final de instalación
**Fuente: Elaboración propia.**

Tras completar la instalación, podemos iniciarlo haciendo clic sobre el icono que ha sido creado en el escritorio.

### 8.1.2. Instalar paquete BiblioMetrics desde GitHub

Una vez se ha abierto la aplicación ”RGui”, para descargar el paquete BiblioMetrics, se hará desde GitHub, ya que el paquete no ha sido subido al repositorio CRAN. Para instalarlo de esta manera, se tendrán que ejecutar una serie de comandos. Lo primero que hay que hacer es instalar un paquete adicional llamado `remotes`, este será cargado y se usará para instalar BiblioMetrics.

```
1 install.packages("remotes")
2 remotes::install_github("Juanmijr/bibliometrics")
```

Listado 8.1: Consola para instalar, añadir y usar remotes e instalar BiblioMetrics

Tras ejecutar los dos comandos, se mostrará en la consola lo que se ve en la figura 8.10. Como se puede observar, se busca instalar los paquetes más actualizados posibles. Para ello, introduciremos el valor ”1” en la consola.

```
> remotes::install_github("juanmijr/bibliometrics")
Using GitHub PAT from the git credential store.
Downloading GitHub repo juanmijr/bibliometrics@HEAD
These packages have more recent versions available.
It is recommended to update all of them.
Which would you like to update?

1: All
2: CRAN packages only
3: None
4: curl       (5.2.1  -> 5.2.2 ) [CRAN]
5: reticulate (1.38.0 -> 1.39.0) [CRAN]

Enter one or more numbers, or an empty line to skip updates:
```

Figura 8.10: Captura de instalación paquete BiblioMetrics desde Github
**Fuente: Elaboración propia.**

Una vez que introduzcamos el valor, comenzará la instalación del paquete. Al finalizar la ejecución, el paquete **BiblioMetrics** estará listo para su uso.

### 8.1.3. Introducción de datos corporativos Universidad de Jaén

Para obtener las métricas necesarias para el análisis de un artículo en Scopus, será necesario utilizar las credenciales de la Universidad de Jaén. Estas credenciales se almacenarán en un archivo, de manera que una vez que las introduzcas, no tendrás que volver a ingresarlas, y el proceso funcionará automáticamente.

En la versión de Windows, el archivo `.Renviron` debe ubicarse en la carpeta `Documentos`, ya que, por defecto, R busca este archivo en dicha ubicación. En ella si no existe, de-

bemos crear un archivo ’’**.Renviron”**, diseñado para almacenar variables de entorno utilizadas en sesiones de R. Este archivo debemos abrirlo con cualquier editor de textos y añadirle lo mostrado en la figura 8.11.

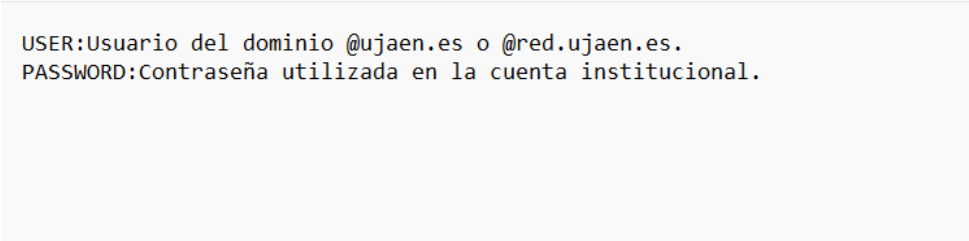



Figura 8.11: Captura de pantalla del archivo .Renviromen
**Fuente: Elaboración propia.**

## 8.2. Apéndice B. Manual de uso del paquete desde consola

Una vez completada la instalación del paquete, deberemos llamar al método `runGUI()` (ver listado 8.2) para que se inicie la interfaz gráfica. Este método abrirá tu navegador predeterminado y ejecutará la interfaz gráfica, mostrando la pantalla de inicio.

```
1 bibliometrics::runGUI()
```

Listado 8.2: Consola para correr la interfaz gráfica de BiblioMetrics

A continuación, explicaremos detenidamente cómo realizar cada una de las funciones que ofrece el paquete BiblioMetrics.

### 8.2.1. Búsqueda y análisis de artículos en Scopus

Primero, una vez en la pantalla principal de la interfaz gráfica del paquete, se puede realizar una búsqueda, ya sea por artículo, por autores o por revistas. En esta subsección, se procederá a realizar una búsqueda de artículos. Para ello, en el primer elemento *select*, se debe seleccionar ”Búsqueda por artículo”. Dado que solo hay una base de datos disponible, se elegirá Scopus. Finalmente, se introducirá la consulta deseada en el campo de texto. Tras esto, se habilitará el botón ”Buscar”, el cual se podrá clicar para obtener los resultados mostrados en la figura 8.12.

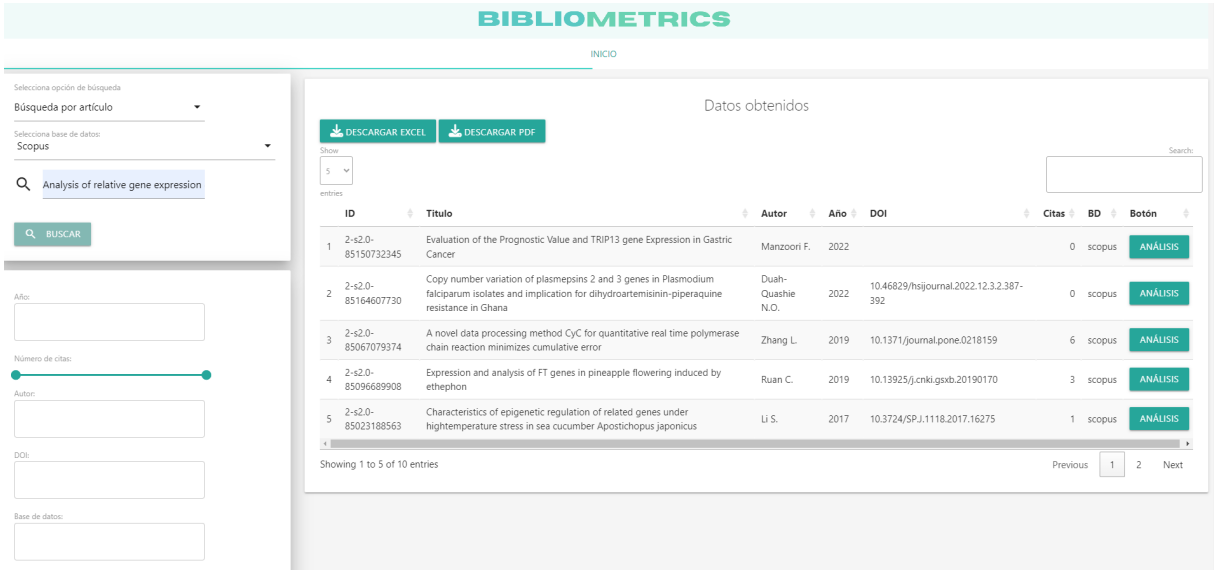



Figura 8.12: Captura de búsqueda de artículos de Scopus
**Fuente: Elaboración propia.**

Una vez obtenida la consulta de artículos, se puede proceder a analizar los datos del artículo. Para ello, se debe pulsar el botón de análisis. Esto habilitará una pestaña llamada ”Análisis” al lado de la pestaña ”Inicio”. Al seleccionarla, se mostrará un análisis exhaustivo del artículo.

### 8.2.2. Búsqueda y análisis de autores

Para obtener información sobre autores, se dispone de diferentes bases de datos, como Scopus y Google Scholar. Se debe realizar una consulta, tal como se ha explicado en el apartado anterior, con la opción de seleccionar una de las dos bases de datos o ambas. La respuesta a la consulta proporcionará los autores listados y permitirá obtener un análisis exhaustivo de cada uno. Se generarán dos tipos de análisis diferentes, dependiendo de si la base de datos utilizada es Google Scholar (Figura 8.14) o Scopus (Figura 8.13).

### 8.2.3. Búsqueda y análisis de revistas en Scopus

Otra opción disponible en el paquete BiblioMetrics es realizar una búsqueda por revistas. Esta consulta se limita a una única base de datos, que es Scopus. Además, es posible llevar a cabo un análisis exhaustivo de cada revista, incluyendo la generación de gráficos interactivos (Figura 8.15).

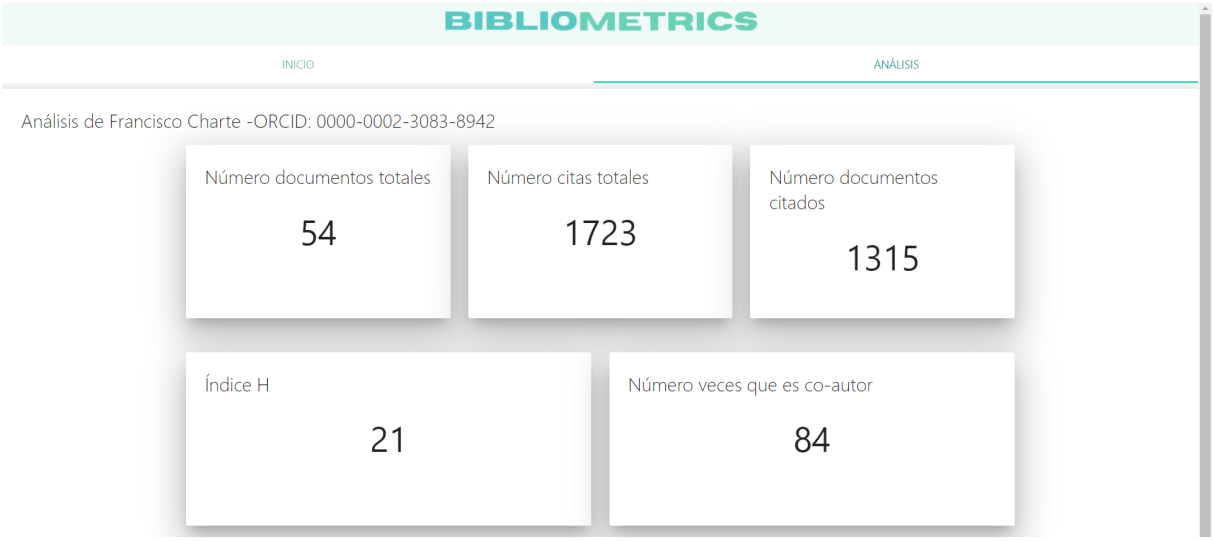



Figura 8.13: Captura de análisis de autor Scopus
**Fuente: Elaboración propia.**

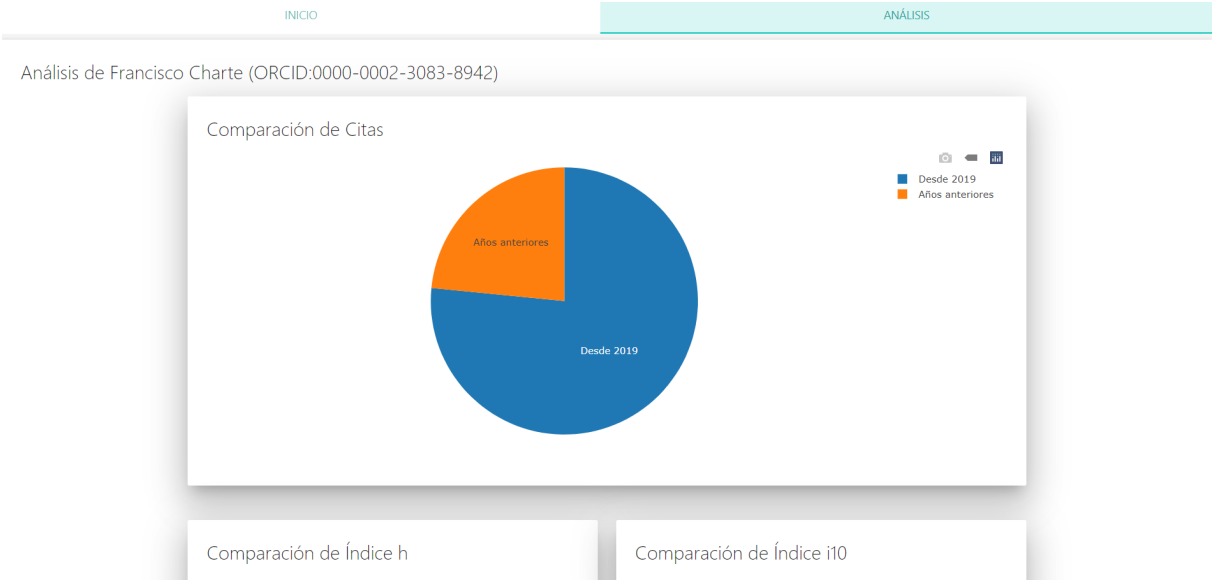



Figura 8.14: Captura de análisis de autor Google Scholar
**Fuente: Elaboración propia.**

### 8.2.4. Exportación de datos

Tras realizar cualquier tipo de búsqueda, como se muestra en la figura 8.12, se presentan dos botones: uno para exportar datos en formato PDF (Figura 8.16) y otro para exportar en formato XLSX (Figura 8.17). Al hacer clic en cualquiera de estos botones, los datos se exportan y formatean de acuerdo con la extensión seleccionada.

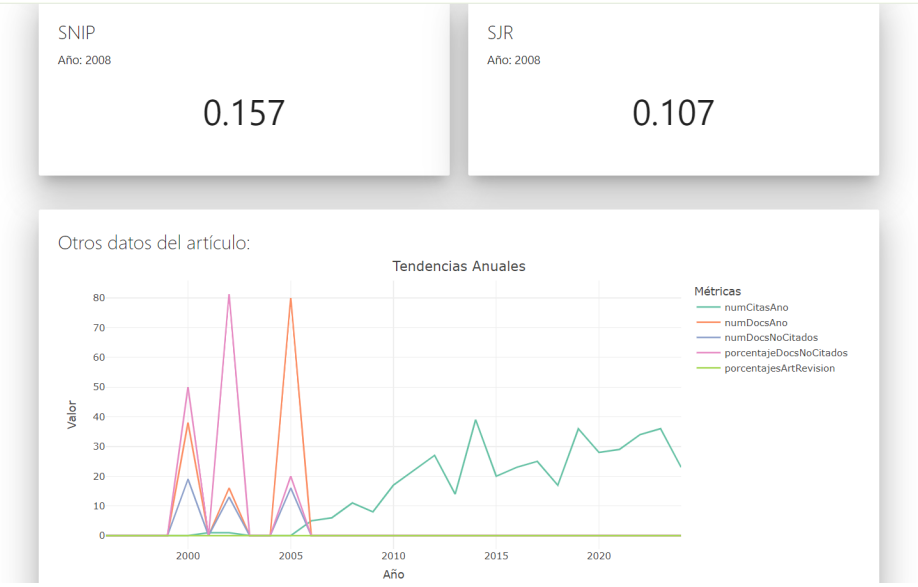



Figura 8.15: Captura de análisis de revista de Scopus
**Fuente: Elaboración propia.**

| ID | Nombre | Apellido | NumDocumentos | Afiliación | Campos | BD |
|---|---|---|---|---|---|---|
| 9-s2.0-40661023100 | Francisco Charte | Ojeda | 54.0 | Universidad de Jaén | Computer Science , Mathematics , Neuroscience | scopus |
| 9-s2.0-57053450500 | D. | Charte | 14.0 | Universidad de Granada | Computer Science , Mathematics , Medicine | scopus |
| 9-s2.0-6508012925 | Ángel | Charte | 11.0 | Universitat de Barcelona | Medicine , Biochemistry, Genetics and Molecular Biology , Neuroscience | scopus |
| 9-s2.0-6506974476 | A. | Charte González | 2.0 | Hospital Universitari Dexeus | Medicine , Health Professions | scopus |
| 9-s2.0-57191824622 | | | 2.0 | Instituto Pirenaico de Ecología | Environmental Science , Agricultural and Biological Sciences | scopus |
| 9-s2.0-6508198744 | | | 1.0 | Hospital San Jorge | Medicine | scopus |
| 9-s2.0-6504611160 | | | 1.0 | Universidad de Zaragoza, Facultad de Medicina | None | scopus |
| 9-s2.0-58125457200 | | | 1.0 | None | Medicine | scopus |
| 9-s2.0-51863127500 | | | 1.0 | Hospital San Jorge | Health Professions | scopus |

Figura 8.16: Captura de una consulta exportada a PDF
**Fuente: Elaboración propia.**

| ID | Nombre | Apellido | nDocumer | Afiliación | Campos | BD |
|---|---|---|---|---|---|---|
| 9-s2.0-406 | Francisco ( | Ojeda | 54 | Universida | Computer | scopus |
| 9-s2.0-570 | D. | Charte | 14 | Universida | Computer | scopus |
| 9-s2.0-650 | Ángel | Charte | 11 | Universita | Medicine , | scopus |
| 9-s2.0-650 | A. | Charte Go | 2 | Hospital U | Medicine , | scopus |
| 9-s2.0-57191824622 | | | 2 | Instituto P | Environme | scopus |
| 9-s2.0-6508198744 | | | 1 | Hospital Sa | Medicine | scopus |
| 9-s2.0-6504611160 | | | 1 | Universidad de Zarag | | scopus |
| 9-s2.0-58125457200 | | | 1 | | Medicine | scopus |
| 9-s2.0-51863127500 | | | 1 | Hospital Sa | Health Pro | scopus |

Figura 8.17: Captura de una consulta exportada a XLSX
**Fuente: Elaboración propia.**

# Bibliografía


[1] Dimitrije Curcic. Number of academic papers published per year, 2023. URL https://wordsrated.com/number-of-academic-papers-published-per-year/.

[2] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill Higher, 2010. ISBN 9780073375977. doi: 0073375977.

[3] Ángel Aguilera García L. Alfonso Ureña López. Tema 2. modelos del proceso, 2020. Transparencias de la asignatura de Fundamentos de Ingeniería Software.

[4] La metodología agile en santander: así ha revolucionado nuestra forma de trabajar, 2021. URL https://n9.cl/t9lmyf. Comprobado en 08-04-2024.

[5] Lasa Gómez Carmen las Heras del Dedo Rafael de, Álvarez García Alonso. *Métodos Ágiles. Scrum, Kanban, Lean*. Anaya Multimedia, 2018. ISBN 8441537712. doi: 9788441537712.

[6] Resolución de 13 de julio de 2023, de la dirección general de trabajo, por la que se registra y publica el xviii convenio colectivo estatal de empresas de consultoría, tecnologías de la información y estudios de mercado y de la opinión pública., 2023. URL https://www.boe.es/eli/es/res/2023/07/13/(5). Comprobado en 12-04-2024.

[7] Resolución de 27 de marzo de 2024, de la dirección general de trabajo, por la que se registra y publica el xviii convenio colectivo estatal de empresas de consultoría, tecnologías de la información y estudios de mercado y de la opinión pública., 2024. URL https://www.boe.es/eli/es/res/2024/03/27/(3). Comprobado en 12-04-2024.

[8] Ángel Aguilera García L. Alfonso Ureña López. Tema 6. modelado de la dinámica, 2020. Transparencias de la asignatura de Fundamentos de Ingeniería Software.

[9] M. García Vega. Apuntes interacción persona-ordenador, 2023. Transparencias de la asignatura de Interacción Persona-Ordenador.

[10] Desconocido. Snip (source normalized impact per paper), . URL https://biblioguias.biblioteca.deusto.es/c.php?g=155487&p=1099289. Visitado el 24/08/2024).

[11] Desconocido. Sjr. scimago journal & country rank: Scimago journal rank, . URL https://biblioguias.biblioteca.deusto.es/c.php?g=515641&p=3525056. Visitado el 24/08/2024).