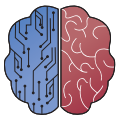
UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería de la Salud



INGENIERÍA
DE LA SALUD

TFG del Grado en Ingeniería de la Salud

Clinical Report Scraping Documentación Técnica

Presentado por Samuel González Martín
en Universidad de Burgos

12 de junio de 2024

Tutores: Antonio Jesús Canepa Oneto – Patricia Saiz López

Índice general

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

Introducción

Se van a empezar los anexos al igual que se inician cualquier proyecto, estudiando si es posible, cuáles serían sus costes para llevarlo a cabo y cuanto se tardaría en conseguir.

Por lo que se ha dividido este anexo en tres subapartados, la planificación temporal, la planificación económica y la viabilidad legal.

A.1. Planificación temporal

En esta sección se detallará el cronograma del proyecto, incluyendo las diferentes etapas del desarrollo, la duración estimada de cada una y los recursos humanos y materiales necesarios para su ejecución.

El método de trabajo seguido ha estado basado en la metodología *Scrum* [APD, 2024]. En este caso, el *Product Owner* sería el equipo de anatomía patológica y el *Scrum Master* sería Jose Antonio Canepa, tutor del TFG.

Se han planteado por lo general sprints de una semana de duración, después de cada sprint se realizaba una tutoría para planificar la siguiente semana. En estas reuniones se hablaba sobre los avances desarrollados la semana previa y se estudiaban los nuevos posibles pasos a tomar en el siguiente sprint. Aproximadamente una vez al mes se organizaban reuniones con Patricia Saiz López, patóloga del HUBU, para mostrar las nuevas actualizaciones del proyecto y tomar nota de las nuevas peticiones solicitadas para el programa.

Tras cada Sprint, marcado como un *issue* en GitHub, se ha hecho un *commit* recogiendo los avances de la semana. Además, también se ha hecho un *commit* cada vez que se conseguía solventar un problema. Esta es la razón por la que en el repositorio de GitHub el número de *commits* es tres veces más que el de *issues*.

Más allá de *commits* e *issues* se han planteado también cinco *milestones* como se puede observar en el diagrama de Gantt A.1. Cada *milestone* representa un objetivo principal en el proyecto. Estos cinco objetivos se corresponden al aprendizaje de las nuevas tecnologías, el desarrollo general de la aplicación, la creación de un registro de pacientes con una base de datos, la consulta de mutaciones usando API clínicas y por último la revisión final de la App.

A continuación se muestra un esquema de la planificación del proyecto:

Aprendizaje

Primer contacto con aplicaciones Shiny y repaso de la programación en R. Fue abierta el 15 de noviembre de 2023 y se cerró el 13 de febrero de 2024. Sus issues asociados son:

- Primer boceto de la app en Shiny. Durante esta issue me centré en aprender las utilidades que me ofrecía Shiny y cuáles podía usar en el TFG.
- Implementar el código en R. En este sprint se implementó en R el código de Lucia Vitores y se adaptó a las nuevas necesidades del TFG.

Desarrollo de la aplicación Shiny

Desarrollo general de la aplicación Shiny e incorporación de las sugerencias solicitadas por Anatomía Patológica. Se abrió el 5 de febrero de 2024 y se cerró el 2 de abril de 2024. Sus issues asociados son:

- Semana 13-20 de febrero. Esta semana se propuso como objetivo finalizar las correcciones del código para poder empezar a programar la visualización de la app.
- Semana 20-27 de febrero. Se avanzó con la aplicación Shiny, se integraron widgets y el emblema del Hospital Universitario de Burgos.

- Semana 27-5 de marzo. Pequeña optimización del código y se empezaron a barajar opciones sobre como incorporar una opción de descarga de archivos a la app.
- Semana 5-12 de marzo. Esta semana se programó la primera reunión con el HUBU, en esta reunión se habló de un nuevo modelo de PDF por lo que se tuvo que adaptar el código a los nuevos cambios. Además, se intentó integrar la idea de sobrescribir archivos CSV para mantener el registro de pacientes en un archivo.
- Semana 13-02 de abril. Además de los cambios en el modelo de los informes, también se solicitó extraer más variables de ellos. A lo largo de esta semana se llevaron a cabo estas solicitudes.

Implementación de los datos de la app en una base de datos

Añadir un historial de pacientes en una base de datos MongoDB que se conecte con la aplicación Shiny. Se abrió el 3 de abril de 2024 y se cerró el 16 de abril de 2024. Sus issues asociados son:

- Semana 02-09 de abril. Esta semana se propuso implementar la base de datos MongoDB al proyecto.
- Semana 09-16 de abril. A inicio de este Sprint se programó una segunda reunión con el HUBU para hablar sobre como organizar y mostrar los datos de los pacientes. También se propuso la idea de las API y se tomó nota de las principales bibliotecas clínicas usadas por los patólogos.

Añadir consultas con API clínicas

Consultar las mutaciones en API clínicas y añadir estos datos a tablas mostradas en pantalla. Se abrió el 16 de abril de 2024 y se cerró el 28 de marzo de 2024. Sus issues asociados son:

- Semana 16-23 de abril. Esta semana se comenzó a programar con la API del NCBI.
- Semana 14-21 de marzo. Se continuó con el trabajo en la API y se añadió una pantalla de inicio en la app.
- Semana 21-28 de marzo. En este sprint se propuso finalizar el código.

Revisión de la App

Revisar que no quede ningún error por solucionar y arreglarlo en caso necesario. Se abrió el 29 de mayo de 2024 y se cerró el 12 de junio de 2024.

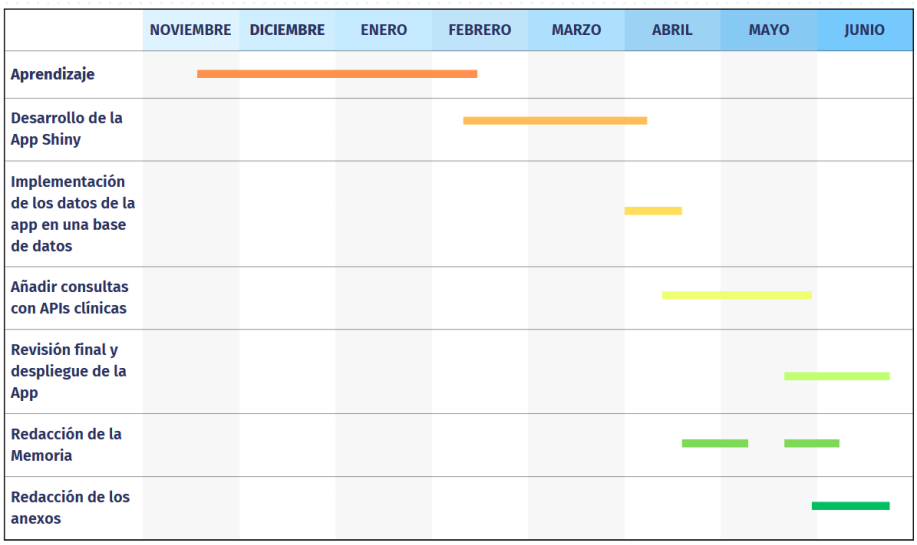


Figura A.1: Diagrama de Gantt

A.2. Planificación económica

Se analizarán los costes asociados al desarrollo del TFG, incluyendo materiales, software, servicios externos y cualquier otro gasto relevante. Se elaborará un presupuesto detallado que permita evaluar la viabilidad económica del proyecto y tomar decisiones informadas sobre la asignación de recursos.

Coste de personal

Este proyecto ha sido llevado a cabo por un desarrollador de software durante un periodo de 7 meses con un contrato de media jornada. Un desarrollador de Software en España gana de media 1780 euros netos al mes, pero hay que tener en cuenta las retenciones del IRPF y la seguridad social.

Concepto	Coste
Salario mensual bruto	1780€
IRPF (19 %)	338,2€
Seguridad Social (28,3 %)	503,74€
Salario bruto mensual	2621,92€
Total por 7 meses	18.353,44€

Coste de Software

Para la programación, todas las herramientas empleadas para son de uso gratuito, desde los paquetes de R y la IDE de RStudio hasta los servicios externos a esta como MongoDB y la API del NCBI. El único gasto a mencionar sería el software Oncomine Reporter usado por el HUBU.

Concepto	Coste
Software Oncomine Reporter	78,50€
Impuestos	16,49€
Total	94,99€

Otros gastos

Todo el trabajo ha sido realizado desde un portátil Asus FX505DT-BQ624. Pero este portátil fue comprado al iniciar la carrera, por lo que es necesario calcularle la respectiva amortización, teniendo en cuenta que tiene una vida útil de 5 años.

Concepto	Coste en €
Ordenador Portatil	750€
Amortización anual	150€
Total	150

A.3. Viabilidad legal

En este apartado hablaremos tanto de los aspectos legales que afectan a este trabajo. Se tratarán tanto de las herramientas informáticas empleadas como de los datos cedidos por el HUBU con los que se ha trabajado.

Herramientas de software

En lo que respecta a las tecnologías empleadas, todas ellas son de software libre, lo que implica que, desde un punto de vista legal, no infringe ninguna ley o licencia de uso.

Datos de pacientes

Los datos sensibles de pacientes, como información médica confidencial o personal identificable, están protegidos por leyes de privacidad y seguridad en muchas jurisdicciones. Algunas de las leyes más conocidas incluyen la Ley de Portabilidad y Responsabilidad del Seguro Médico (HIPAA) en los Estados Unidos y el Reglamento General de Protección de Datos (GDPR) en la Unión Europea.

Para compartir datos sensibles de pacientes de forma legal con terceros, es fundamental cumplir con las regulaciones de privacidad de datos aplicables. Esto puede implicar obtener el consentimiento explícito del paciente para compartir su información, garantizar que se sigan los protocolos de seguridad adecuados para proteger los datos durante la transmisión y el almacenamiento, y establecer acuerdos de confidencialidad sólidos con los terceros involucrados

En este caso se ha optado por utilizar métodos de pseudoanonimización para compartir datos de pacientes, de manera que no puedan ser fácilmente identificados por terceros. No obstante, tanto la aplicación Shiny como la base de datos de MongoDB son ejecutadas de forma local, lo que implica que tanto los software como los datos no tienen conexión con ninguna red exterior a la propia computadora que los aloje. Por lo tanto, todos los datos compartidos en la app no son compartidos con terceros no autorizados.

Apéndice B

Documentación de usuario

B.1. Requisitos software y hardware para ejecutar el proyecto.

Para ejecutar este software primero de todo se necesitará tener instalado R y cualquier IDE que permita manipular archivos en R. En este caso se ha usado RStudio, pero cualquier otra es viable (Visual Studio Code, TinnR, Eclipse, Rbase, ...). El proyecto se ha trabajado en Windows, por lo tanto, las instrucciones de instalación están dirigidas a este sistema operativo en concreto.

Por separado, también deberemos instalar el controlador de MongoDB, MongoDB Compass, para que el sistema pueda crear los registros de los pacientes de forma local. Como instalar todos estos programas viene detallado en el siguiente apartado.

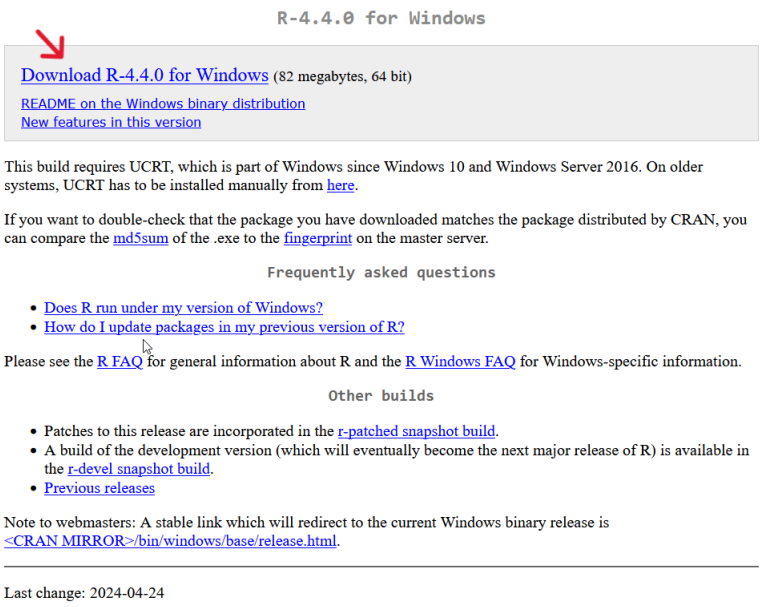
B.2. Instalación / Puesta en marcha

Instalación de R

Para instalar R en Windows, se puede descargar desde la página de CRAN [CRAN, 2024]. Una vez en la página, debe hacerse clic en el apartado "*Download*", como se muestra en la imagen B.1. Esto descargará un archivo ejecutable (.exe) que se encontrará en la carpeta de descargas.

Al ejecutarlo, se debe seleccionar el idioma de instalación B.2. Una vez seleccionado, aparecerán las licencias B.3 y se ofrecerá la opción de elegir el directorio de instalación B.4; en este caso, se mantendrá la carpeta por

defecto. Luego, se seleccionarán los componentes a instalar y se especificará si se desea utilizar las opciones de configuración, como se muestra en las imágenes B.5 y B.6. Por último, antes de comenzar la instalación, se elegirá una carpeta para guardar los accesos directos del programa B.7 y se definirán las tareas adicionales que se realizarán durante la instalación B.8.



R-4.4.0 for Windows

Download R-4.4.0 for Windows (82 megabytes, 64 bit)

README on the Windows binary distribution
New features in this version

This build requires UCRT, which is part of Windows since Windows 10 and Windows Server 2016. On older systems, UCRT has to be installed manually from here.

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the md5sum of the .exe to the fingerprint on the master server.

Frequently asked questions

- Does R run under my version of Windows?
- How do I update packages in my previous version of R?

Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information.

Other builds

- Patches to this release are incorporated in the r-patched snapshot build.
- A build of the development version (which will eventually become the next major release of R) is available in the r-devel snapshot build.
- Previous releases

Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.html.

Last change: 2024-04-24

Figura B.1: Descargar R

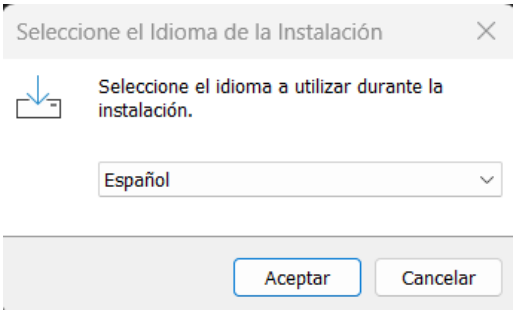


Figura B.2: Selección de idioma

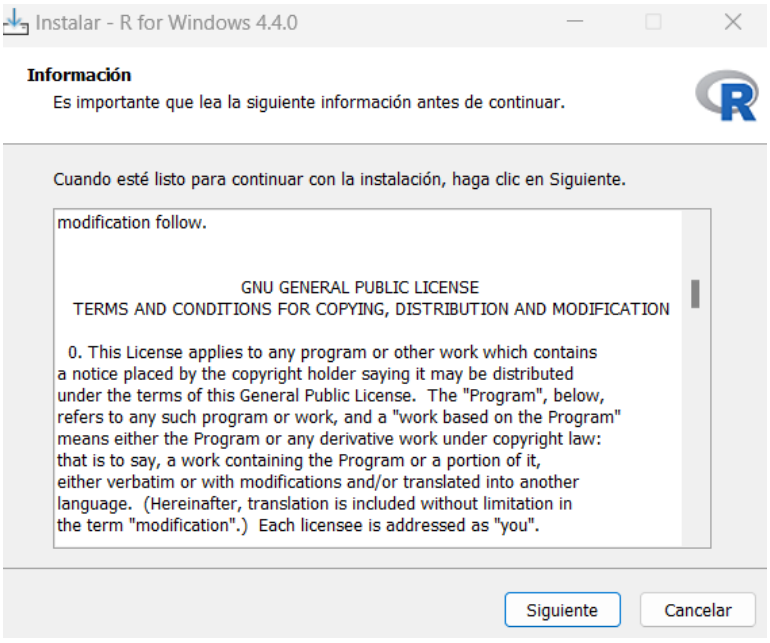


Figura B.3: Licencias

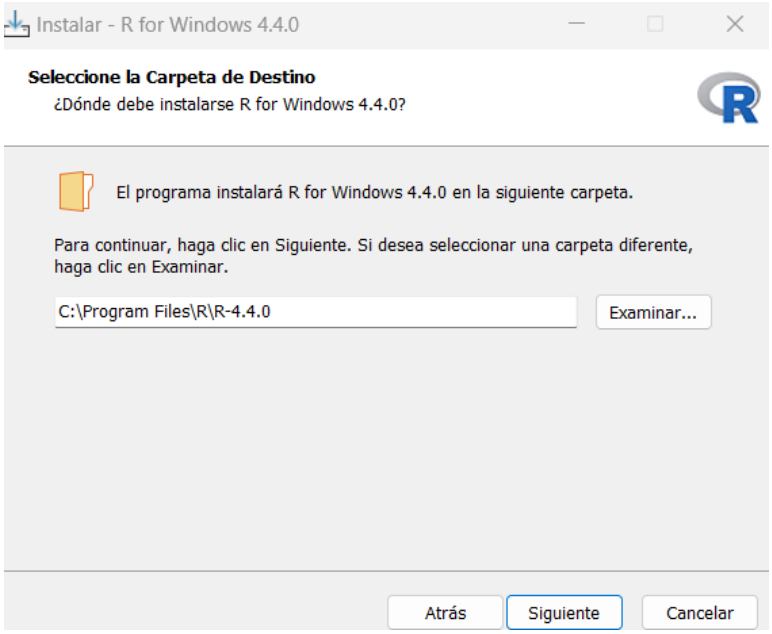


Figura B.4: Selección de carpeta de destino

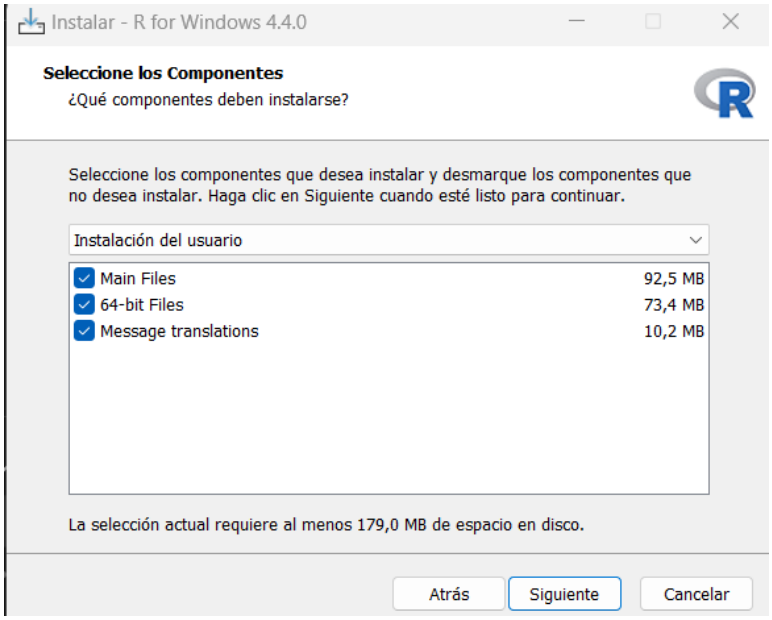


Figura B.5: Selección de componentes

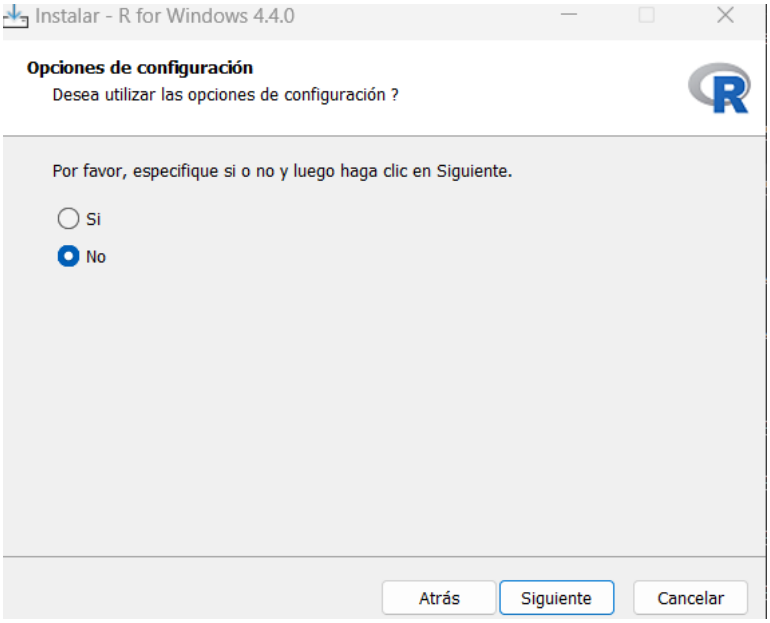


Figura B.6: Opciones de configuración

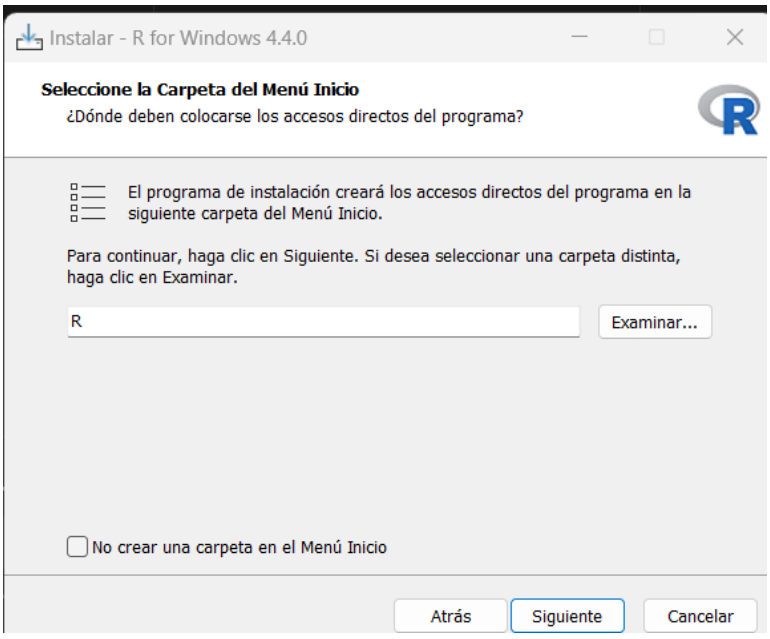


Figura B.7: Elección de carpeta para los accesos directos

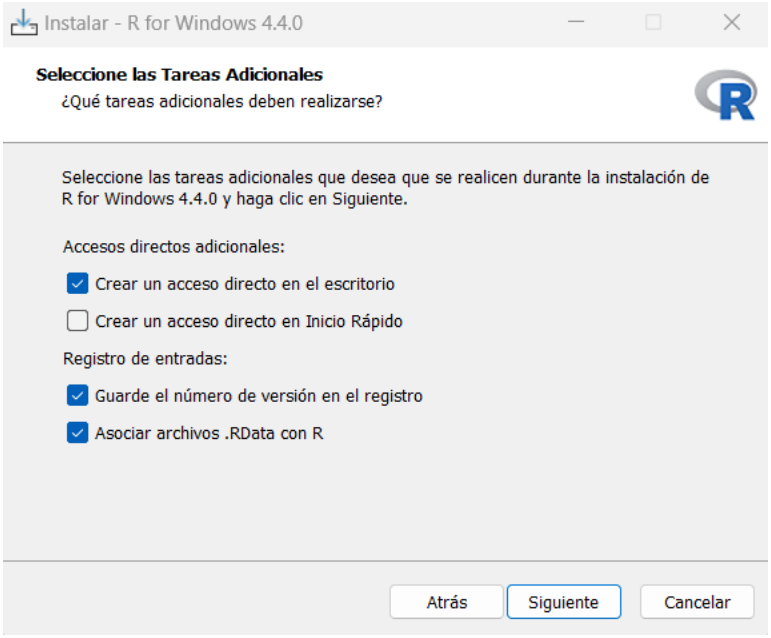


Figura B.8: Tareas adicionales

Instalación de RStudio

La implementación de RStudio es similar a la de R. Para descargar el archivo B.9 de instalación hay que acceder a la web Posit [Posit, 2024].

Al ejecutar el archivo se nos solicitará indicar el directorio de destino B.4 y la carpeta para los accesos directos de RStudio B.11. Una vez cumplimentada esta información comenzará la instalación.

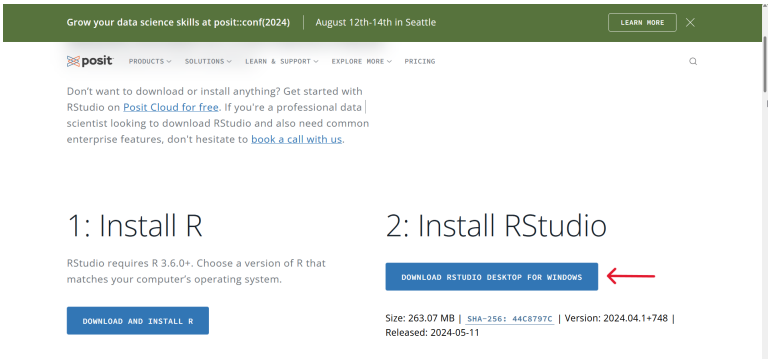


Figura B.9: Descargar archivo de instalación

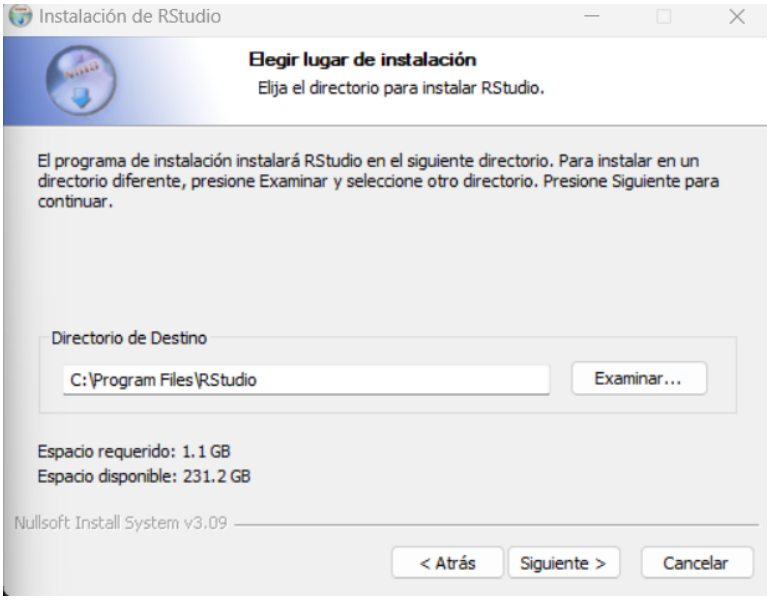


Figura B.10: Selección de carpeta de destino

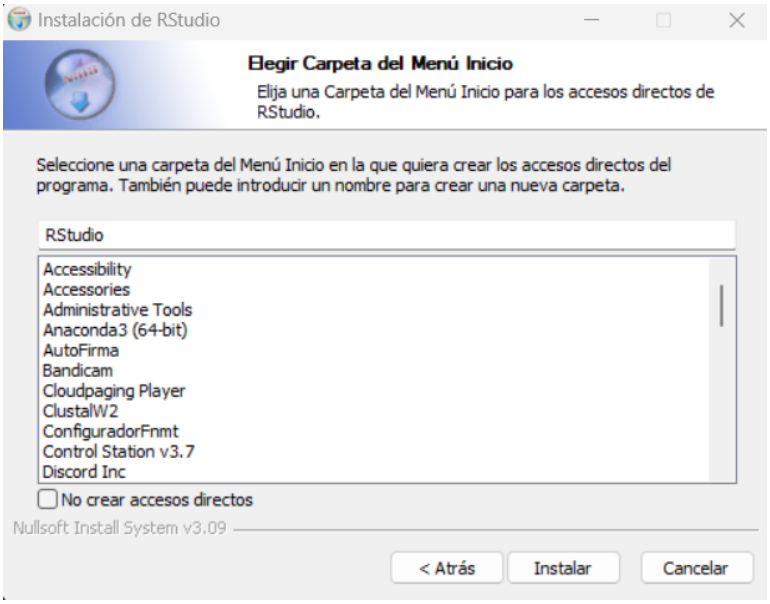


Figura B.11: Elección de carpeta para los accesos directos

Instalación de MongoDB Compass

En la página oficial de MongoDB [MongoDB, 2024] podemos encontrar el instalador de MongoDB Compass B.12. En este caso no deberemos configurar nada, al ejecutar el archivo descargado se instalará la aplicación automáticamente.

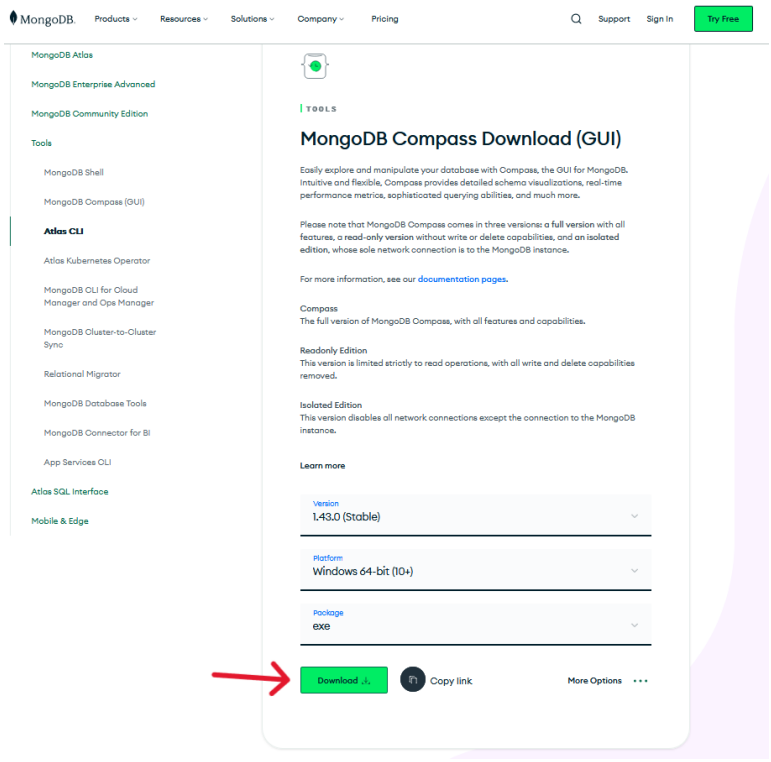


Figura B.12: Descarga instalador de MongoDB Compass

Por último, para terminar de poner en marcha la aplicación Shiny solo deberemos clonar el repositorio de GitHub, abrir el archivo PDF-Scrapping.R y ejecutarlo.

Las instrucciones para clonar un repositorio son [GitHub, 2024]:

- Entrar en la página de GitHub del repositorio a clonar.
- Pulsar el botón verde nombrado como "<> Code" B.13.
- Seleccionar el modo de clonación (HTTPS, SSH o GitHub CLI).
- Copiar el URL que nos aparezca.

- Abrimos Git Bash y nos ubicamos en el directorio donde queremos clonar el repositorio.
- Escribimos “git clone” y seguida la url copiada.

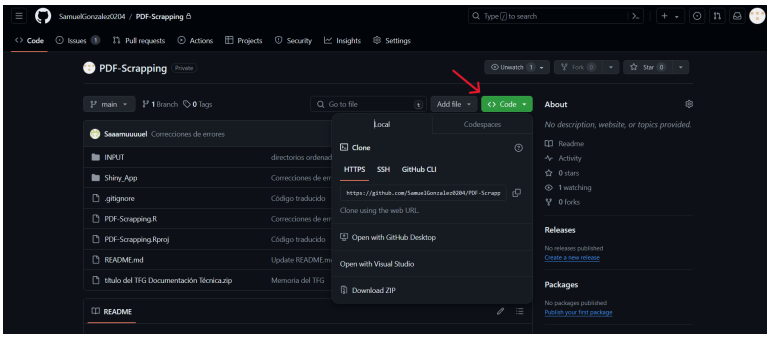


Figura B.13: Clonación de repositorio

B.3. Manuales y/o Demostraciones prácticas

Siguiendo el diagrama de usos del anexo F se van a mostrar unas demostraciones de cada uno de sus usos.

Procesar PDF

En la figura B.14 vemos como se han extraído los datos de 3 archivos subidos.

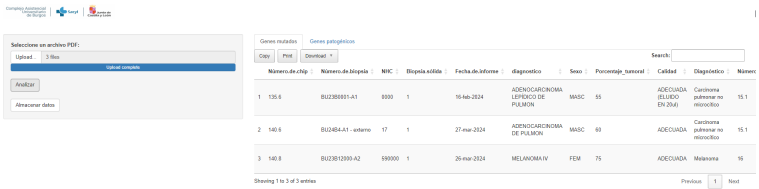


Figura B.14: Demostración de procesar PDF

Consultar búsquedas en bases de datos clínicas

La columna Patogenicidad.Buscada de la tabla de la figura B.15 corresponde a las búsquedas de las mutaciones en la base de datos de ClinVar.

Figura B.15: Demostración de consulta en bases de datos clínicas

Modificar tablas

Al pulsar dos veces sobre cualquier casilla de las tablas se abrirá un editor con el que podremos modificar el valor de dicha casilla.

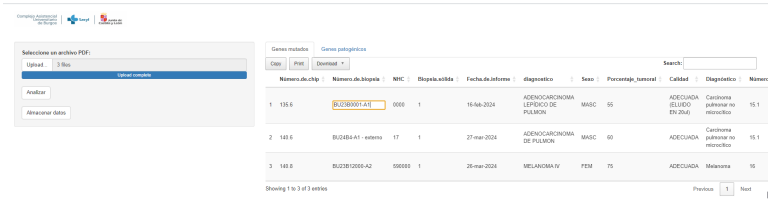


Figura B.16: Demostración de modificación de tablas

Ordenar valores

En este ejemplo B.17 se ha ordenado la columna Porcentaje_tumoral en orden descendente

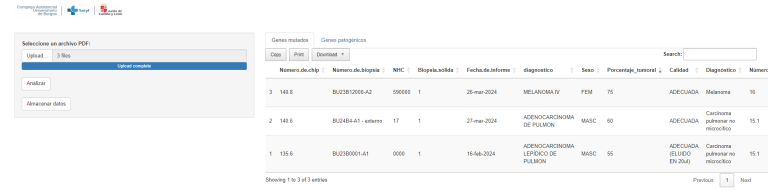


Figura B.17: Demostración de ordenar valores

Filtrar filas

En la imagen B.18 se han buscado solo aquellos registros que sean de un paciente masculino.

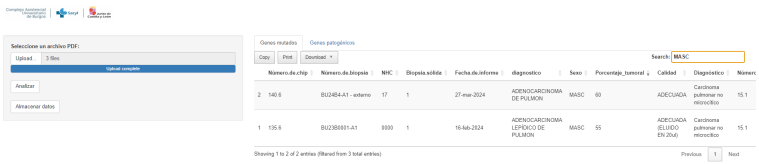


Figura B.18: Demostración de filtrar filas

Exportar datos

En este ejemplo se han extraído los datos en un documento CSV B.19 y se muestra la visualización del archivo descargado B.20.

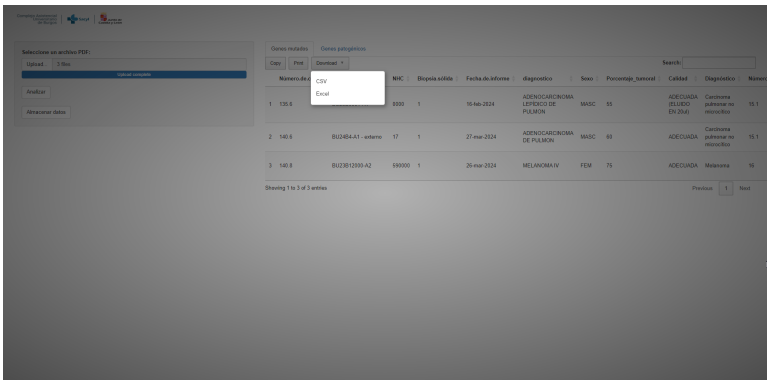
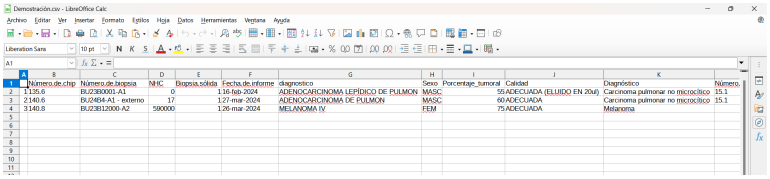


Figura B.19: Demostración de exportar datos



Número.de.chip	Número.de.biopsia	NHC	Biopsia.sólida	Fecha.de.informe	diagnostico	Sexo	Porcentaje_tumoral	Calidad	Diagnóstico	Número.
1 135.6	BU23B0001-A1	0	1	16-feb-2024	ADENOCARCINOMA LEPÍDICO DE PULMON	MASC	55	ADECUADA (ELUIDO EN 20ul)	Carcinoma pulmonar no microcítico	15.1
2 140.6	BU24B4-A1 - externo	17	1	27-mar-2024	ADENOCARCINOMA DE PULMON	MASC	60	ADECUADA	Carcinoma pulmonar no microcítico	15.1
3 140.8	BU23B12000-A2	590000		26-mar-2024	MELANOMA IV	FEM	75	ADECUADA	Melanoma	

Figura B.20: Demostración de exportar datos

Crear historial de pacientes

En la imagen B.21 se muestran los registros de los documentos analizados en la aplicación MongoDB Compass.

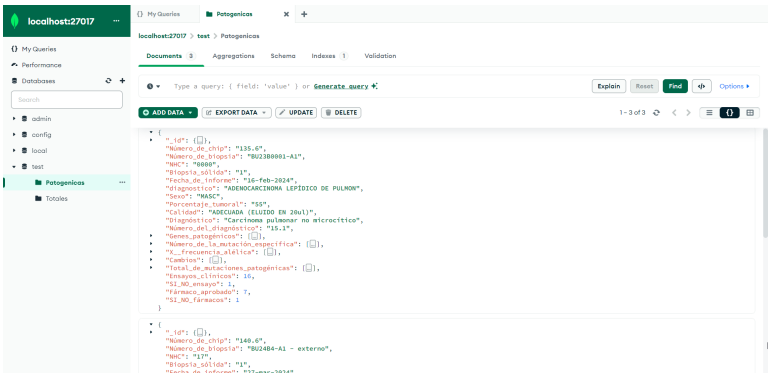


Figura B.21: Demostración de crear historial de pacientes

Apéndice C

Manual del programador.

C.1. Estructura de directorios

A continuación se procede a describir cada uno de los documentos que podemos encontrar en el repositorio de GitHub y en que carpeta podemos encontrarlos siguiendo el esquema C.1.

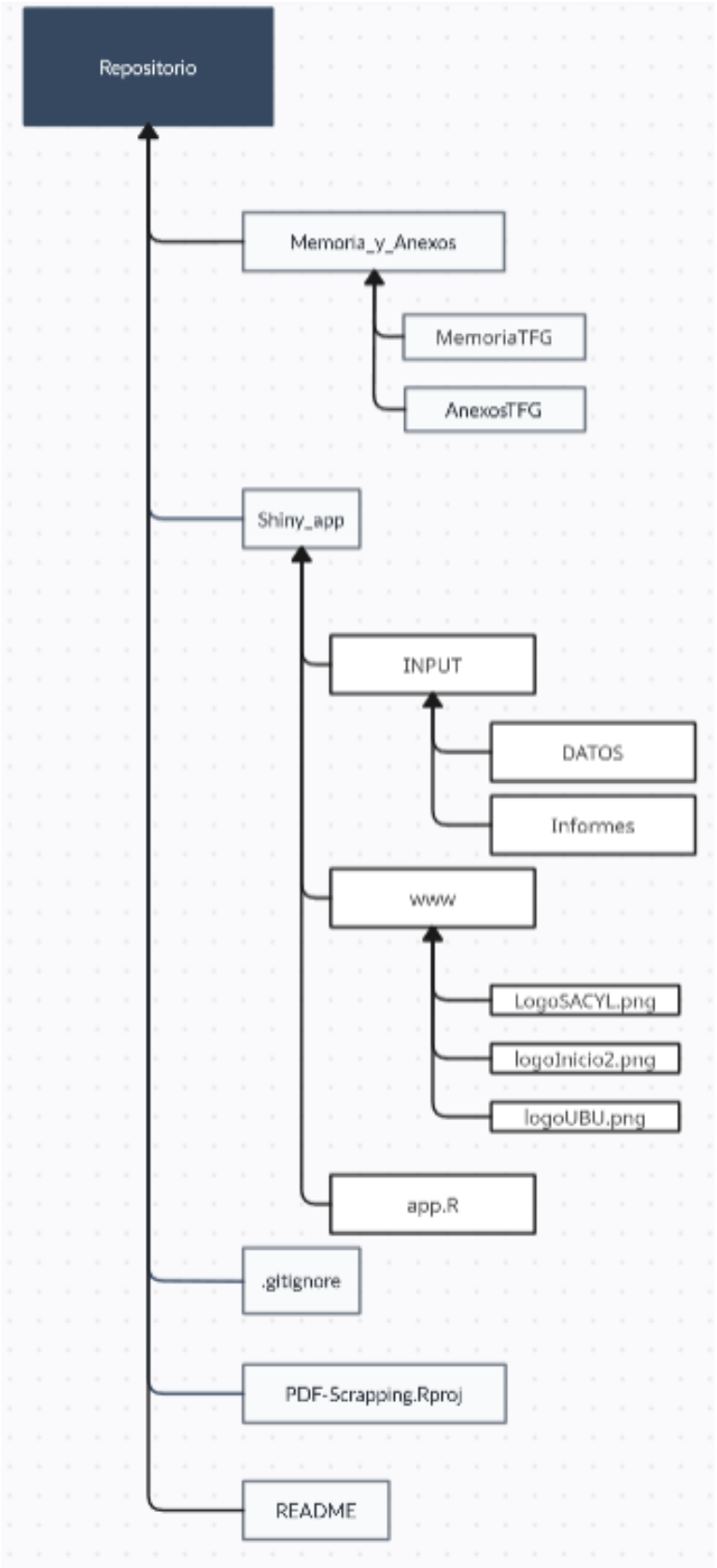


Figura C.1: Diagrama de estructura de directorios

Repositorio

En la pantalla inicial del repositorio podemos encontrar los siguientes documentos:

- Memoria_y_Anexos: carpeta con la documentación relevante al TFG.
- Shiny_App: carpeta con todos los documentos empleados en el código de la app, incluido el propio código.
- .gitignore: archivo que lista patrones de archivos y directorios que Git debe ignorar.
- PDF-Scrapping.R: archivo de R que contiene el ejecutable de la aplicación y a su vez ha sido usado como cuaderno de pruebas.
- PDF-Scrapping.Rproj: archivo de proyecto de RStudio. Contiene las configuraciones específicas del proyecto y permite la conexión con Git.
- README.md: archivo de texto en formato Markdown que contiene una introducción al proyecto.

Carpeta Memoria_y_Anexos

Dentro de esta carpeta encontramos dos archivos:

- MemoriaTFG: documento PDF que contiene la memoria del TFG.
- AnexosTFG: documento PDF con los anexos complementarios a la memoria.

Carpeta Shiny_App

Dentro de esta carpeta tenemos los siguientes archivos:

- INPUT: carpeta con los documentos PDF y XLSX usados por la aplicación
- www: capeta con las imágenes cargadas en la interfaz.
- app.R: archivo de R con toda las líneas de programación del trabajo.

Carpeta INPUT

Los archivos dentro de esta carpeta son:

- DATOS: carpeta con los documentos XLSX.
- Informes: carpeta con los documentos PDF.

Carpeta DATOS

Los respectivos documentos XLSX que nos encontramos en este directorio son:

- Diagnostico.xlsx: tabla con los posibles diagnósticos y su numeración asociada.
- Genes.xlsx: tabla con los genes del panel de genes estudiado y su numeración asociada.

Carpeta Informes

En esta carpeta nos encontramos todos los informes clínicos anonimizados cedidos por el HUBU.

Carpeta www

Las imágenes que nos encontramos en esta carpeta son:

- LogoSACYL.png: imagen del emblema de la Sanidad de Castilla y León.
- logoInicio2.png: imagen de la pantalla de inicio.
- logoUBU.png: imagen del emblema de la Universidad de Burgos.

C.2. Compilación, instalación y ejecución del proyecto

La compilación del proyecto se desarrolla igual que en anexo B.2. A nivel de programación también es relevante tener en cuenta la instalación de los paquetes indicados en la sección de metodología de la memoria. Para la instalación de estos paquetes bastará con ejecutar las líneas mostradas en la

```
install.packages(c("pdftools", "tidyverse", "readxl", "xlsx", "mongolite", "rentrez", "shiny"))
install.packages("DT", repos = "http://cran.us.r-project.org")
```

Figura C.2: Instalación de paquetes

imagen C.2 y que podemos encontrar en el archivo PDF-Scrapping.R. Otro aspecto a contemplar es que para acceder a la base de datos de MongoDB se debe ejecutar el comando "mongod" en el símbolo del sistema del ordenador.

C.3. Instrucciones para la modificación o mejora del proyecto.

A continuación se van a profundizar más en las líneas futuras nombradas en la memoria.

Consulta en más bases de datos genómicas

Como se mencionó anteriormente, el software que genera los documentos PDF tiene limitaciones para determinar la afectación de ciertas mutaciones. Estas mutaciones deben ser buscadas posteriormente por los patólogos responsables en diferentes bases de datos genómicas para completar la información faltante. Entre las más utilizadas se encuentran OncoKB, Varsome, ClinVar, Cbioportal y Franklin. De estas, ClinVar es la única que se ha podido incorporar hasta el momento.

Aunque el programa desarrollado no puede consultar ninguna de las otras cuatro bibliotecas biomédicas, es posible integrar cualquiera de ellas en el código. En esencia, esto implicaría seguir la estructura utilizada para ClinVar.

El único aspecto que podría complicar este proceso es la obtención del permiso para usar la API de cualquiera de estas bibliotecas. Para conseguir este permiso, se debe completar un cuestionario sobre la institución que utilizará la API y el propósito de su uso.

Gestión de usuarios

Tanto la aplicación Shiny como la base de datos MongoDB son accesibles para cualquier persona con acceso al ordenador donde están instaladas o a cualquier otro ordenador que comparta la conexión a la base de datos.

Para evitar problemas y proteger la información en caso de accesos no autorizados, sería recomendable desarrollar un sistema de gestión de usuarios. En MongoDB, esto se puede lograr configurando usuarios y contraseñas para la base de datos específica. Simplemente, selecciona la base de datos y en la pestaña "*Authentication*" introduce las credenciales deseadas.

En el caso de la aplicación Shiny, el proceso es un poco más complejo. Se debe configurar una nueva base de datos para almacenar las credenciales de los usuarios registrados. Luego, utilizando el paquete shinyFeedback, se puede validar el ingreso de los usuarios. Dentro de la función server(), se llaman las funciones feedback(), feedbackWarning(), feedbackDanger() o feedbackSuccess() según corresponda, para verificar si los datos de ingreso coinciden con las entradas en la base de datos.

Para recoger las credenciales del usuario se pueden emplear las funciones de Shiny textInput(), para recoger el nombre del usuario, y passwordInput(), para introducir la contraseña [Wickham, 2020]. Para consultar estos datos en MongoDB, bastaría con cargar la base de datos de MongoDBcompass que contenga las credenciales de los usuarios validados en R como un *dataframe* y se compararía la pareja de datos dados con los registros del *dataframe*.

Mostrar datos estadísticos

Sería una muy buena idea añadir una utilidad a la web que nos muestre datos estadísticos relevantes para los patólogos. Por ejemplo, que mutaciones son más frecuentes en un tipo de cáncer en función del sexo o una gráfica que muestre la frecuencia de aparición de mutaciones en cada gen.

Esta nueva utilidad podría mostrarse en una nueva ventana de nuestra web. La función tabsetPanel() es una posible vía para suplir esta necesidad. Se puede crear un panel para cargar los datos y en el que se muestren otros dos tabsetPanel(), uno para cada tabla, y otro panel par mostrar datos estadísticos en el que con un widget de lista desplegable se pueda seleccionar el tipo de estadística que se quiere visualizar.

Por otro lado, para sacar gráficas de los datos de MongoDB, la mejor opción sería el paquete de Tidyverse llamado Ggplot2 [Rstudio, 2023] que ya hemos comentado en el apartado de metodología. Con este paquete podremos obtener una amplia gama de gráficas como, gráficos de dispersión, histogramas o gráficos de secciones, entre otras, como podemos apreciar en C.3 y C.4.

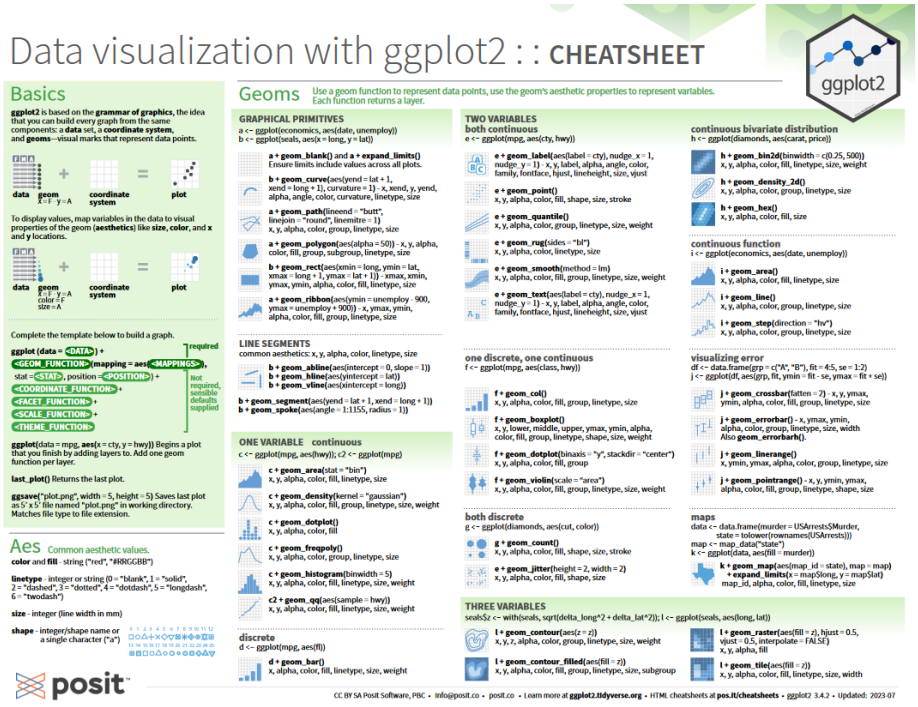


Figura C.3: CheatSheet de ggplot2

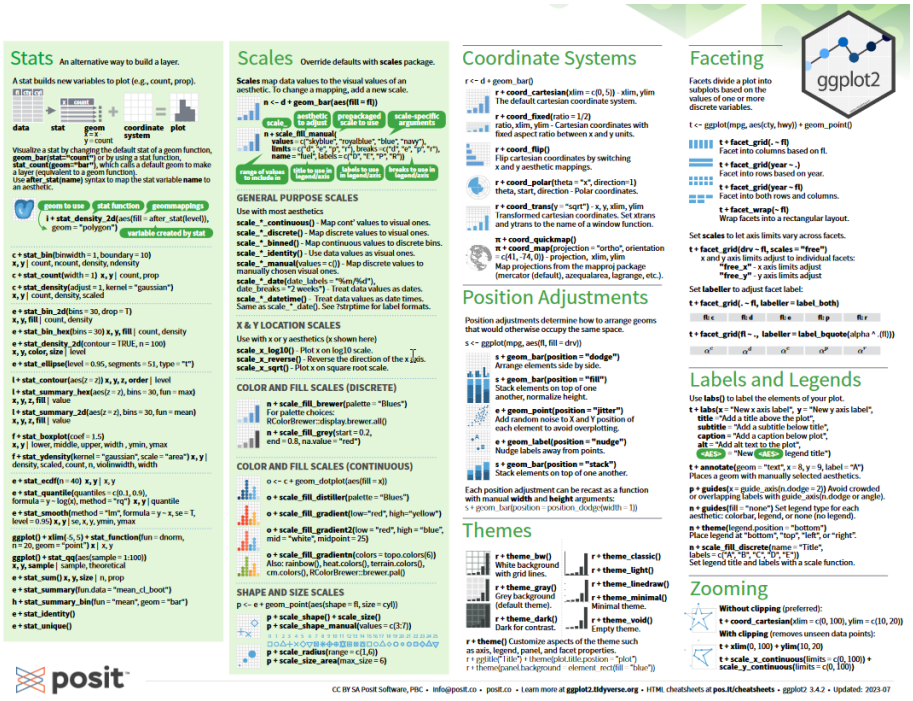


Figura C.4: CheatSheet de ggplot2

Programación funcional

Para optimizar más el código actual es posible usar la biblioteca purrr [Wickham and Henry, 2023] del paquete tidyverse. Purrr permite aplicar funciones de manera más eficiente gracias a su familia de funciones map.

En el caso de que el tamaño de los datos aumente, se podrían aplicar distintos paquetes. En primer lugar, cabe mencionar el paquete data.table [Wickham et al., 2023], una alternativa altamente eficiente a data.frame, que ofrece una sintaxis concisa y operaciones rápidas para filtrar, agregar y transformar datos, reduciendo significativamente los tiempos de ejecución de la aplicación.

Finalmente, para datos a gran escala, se podría considerar el uso de Arrow [Richardson et al., 2024], un paquete que permite trabajar con datos en memoria de manera eficiente y facilita la interacción con otros lenguajes y herramientas de procesamiento de datos a gran escala. Implementar Arrow aseguraría que la aplicación mantenga su rendimiento y capacidad de respuesta, incluso con volúmenes de datos crecientes.

Apéndice D

Descripción de adquisición y tratamiento de datos

D.1. Descripción formal de los datos

En esta sección se realiza una descripción detallada de los datos utilizados para llevar a cabo el proyecto.

El componente principal de entrada con el que se ha trabajado son los documentos en formato PDF. Estos documentos consisten en informes clínicos de pacientes generados a través del software Oncomine Reporter y proporcionados por el departamento de anatomía patológica. Contienen información relevante sobre el paciente, como el sexo, el número de biopsia y las variantes detectadas en la secuencia de ADN.

Previamente, estos informes fueron sometidos a un proceso de pseudoanonimización, el cual implicó la eliminación de datos directamente identificables, como el nombre o el número de historia clínica. Esto garantiza la confidencialidad de los pacientes, pero permite la posibilidad de reidentificarlos mediante claves exclusivas que solo el hospital puede asociar con el paciente en caso de ser necesario.

Además de los documentos PDF, se han empleado dos tablas en formato CSV. La primera tabla D.1 contiene los 52 genes del panel utilizados para la detección de mutaciones en los informes, junto con su respectiva numeración. En la segunda tabla D.2 se encuentran los diversos diagnósticos posibles, también asociados a su número de referencia.

GEN	Número gen
ABL1	1
AKT1	2
AKT3	3
ALK	4
AR	5
AXL	6
BRAF	7
CCND1	8
CDK4	9
CDK6	10
CTNNB1	11
DDR2	12
EGFR	13
ERBB2	14
ERBB3	15
ERBB4	16
ERG	17
ESR1	18
ETV1	19
ETV4	20
ETV5	21
FGFR1	22
FGFR2	23
FGFR3	24
FGFR4	25
GNA11	26
GNAQ	27
HRAS	28
IDH1	29

Figura D.1: CSV con los genes

DIAGNÓSTICO	NÚMERO DIAGNÓSTICO
Carcinoma del tracto biliar	1
Colangiocarcinoma	1.1
Carcinoma de la vesícula biliar	1.2
Cáncer de vejiga	2
Carcinoma vesicular uroterial	2.1
Cáncer de mama	3
Triple negativo de mama	3.1
Neoplasia del sistema nervioso central	4
Glioma	4.1
Cáncer cervical	5
Condrosarcoma	6
Cáncer colorrectal	7
Carcinoma de células escamosas cutáneas	8
Carcinoma endometrial	9
Cáncer esofágico	10
Cáncer gástrico	11
Tumor del estroma gastrointestinal	11.1
Cáncer de cabeza y cuello	12
Cáncer de riñón	13
Cáncer de hígado	14
Carcinoma hepatocelular	14.1
Cáncer de pulmón/pulmonar	15
Carcinoma pulmonar no microcítico	15.1
Carcinoma pulmonar microcítico	15.2
Melanoma	16
Carcinoma de células Merkel	17
Mesotelioma	18
Neoplasia mixto neuroendocrino no- neuroendocrino	19
Carcinoma neuroendocrino	20

Figura D.2: CSV con los diagnósticos

D.2. Descripción clínica de los datos.

Para describir los datos voy a dividir este apartado en dos partes, una para hablar sobre los documentos PDF y la otra sobre los documentos CSV.

PDF

En sus líneas podemos encontrar distintos campos de análisis para el diagnóstico, pero no todas las variables son de igual importancia. Nos vamos a centrar en los datos cuya extracción ha sido solicitada:

Número de chip

Este dato no se obtiene directamente del contenido del PDF, sino que se encuentra en el nombre del documento. Consiste en el número de carrera y, separado por un punto, el número correspondiente al paciente dentro de la carrera. Por ejemplo, el formato sería 140.5, donde 140 representa el número de la carrera y la biblioteca del paciente correspondería al chip número 5 de la placa del Ion Chef.

Número de biopsia

El número de biopsia incluye información detallada como dos letras que identifican la provincia donde se realizó la biopsia, dos números que representan el año, una letra (B para biopsia, P para punción o C para citología) que indica el tipo de muestra, un número de registro secuencial dentro del año y el número de la sección de la muestra total al que pertenece la porción analizada. Por ejemplo, en el caso de BU24B0001-A1, las letras BU hacen referencia a Burgos, el número 24 corresponde al año del informe, B indica que la muestra fue obtenida mediante biopsia, 0001 representa el primer paciente del año y A1 indica la sección de la muestra utilizada para el estudio.

NHC

El número de historial clínico es un identificador único asignado a cada paciente. Este número es usado por el hospital para organizar y acceder a la información clínica del paciente.

Fecha de informe

Como se indica en su nombre, es la fecha en la que se redactó el informe en formato 01-mmm-2024.

Diagnóstico

Esta sección es cumplimentada por el patólogo con su diagnóstico para el paciente.

Sexo

Variable que indica si el paciente es masculino (MASC) o femenino (FEM).

Porcentaje tumoral

Representa cuantas de las células del tejido obtenido en el microcorte de la muestra son tumorales en relación con el total de células de la porción.

Calidad de la muestra

Específica si la muestra estaba en una concentración adecuada antes de ser introducida al Ion Chef o si se ha necesitado diluir para ajustarla a una concentración de 10 ng/ml.

Tipo de cáncer de la muestra

Indica la procedencia de la muestra.

Mutaciones detectadas

Todos los genes del panel analizado en los que se ha detectado mutaciones en la biblioteca genómica del paciente.

Frecuencia del alelo

Porcentaje de apariciones de una determinada variante del gen en relación con todas sus posibles variantes.

Fusiones de genes

Genes que se han fusionado detectados en la secuencia ARN.

Significado clínico de la mutación

Nivel de afectación que tiene una determinada mutación en un paciente contrastado en la base de datos del software de Oncomine Reporter.

Tratamientos disponibles

Número de tratamientos para las mutaciones detectadas. Esta documentación es contrastada en las bases de datos de FDA (Food and Drug Administration), NCCN (National Comprehensive Cancer Network), EMA (European Medicines Agency) y ESMO (European Society for Medical Oncology).

Ensayos clínicos

Número de ensayos en los que se han probado los tratamientos encontrados.

CSV

El primer CSV contiene todos los genes que se encuentran en la placa de 52 genes. De estos 52 genes:

- 35 genes para SNV (*Single Nucleotide Variant*), entre las que se incluyen variantes de múltiples nucleótidos y las inserciones y deleciones.
- 19 genes para el estudio de variantes en el número de copias.
- 23 genes para el estudio de fusiones

Los dos primeros grupos de genes son detectados en la secuencia de ADN, mientras que las fusiones son detectadas en la secuencia de ADNc.

Cada gen está asociado a un número identificativo único. Esta numeración es consultada y mostrada en la app acompañando cada mutación.

Por otro lado, el segundo CSV mantiene la relación entre los diagnósticos y su numeración correspondiente. Cada tejido tiene un número vinculado, si dos lesiones comparten tejido se añade al número un punto y a la derecha de este se comienza otra numeración. Por ejemplo, cáncer de pulmón/pulmonar, carcinoma pulmonar no microcítico y carcinoma pulmonar microcítico son lesiones que afectan a los pulmones, número 15, y como son afectaciones distintas en el mismo tejido se numeran de tal forma que la primera es 15, la segunda es 15.1 y la última 15.2 respectivamente.

D.3. Descripción informática de los datos.

Inicialmente, partimos de documentos PDF que van a ser convertidos a una lista de listas tipo *String*, una lista por cada línea de texto en el documento. Estas líneas son leídas en busca de expresiones lógicas que son indicativo de una variable de interés. Todas las variables son trabajadas como listas de elementos tipo *String* a excepción de:

- Mutaciones detectadas. Consiste en una lista de listas que contienen los genes mutados en cadenas de texto.
- Frecuencia del alelo. Lista de listas con porcentajes en formato *String*.
- Fusiones de genes. Al igual que los anteriores, es una lista de listas con datos en formato de cadena de texto.

- Patogenicidad y patogenicidad buscada. También son guardadas como lista de listas de tipo *String*.
- Número de mutaciones. Es una lista de listas con números asociados a las mutaciones detectadas.
- Ensayos clínicos. Variable de tipo *Integer*.
- Tratamientos disponibles. Variable tipo *Integer*.
- Si/No fármacos y ensayos. Datos tipo *Integer* que solo pueden tomar como valores 1 o 0.

Todos estos valores son unidos en dos *dataFrames* finales que son mostrados en pantalla y enviados a MongoDB para su almacenamiento.

Apéndice E

Especificación de Requisitos

En este anexo se procederá a detallar cuáles son las utilidades y las tareas asociadas a la aplicación. Primero se va a exponer el diagrama de casos de uso y acto seguido se expondrán las tablas de especificaciones.

E.1. Diagrama de casos de uso

En la imagen E.1 podemos observar el diagrama de casos de uso correspondiente a la aplicación.

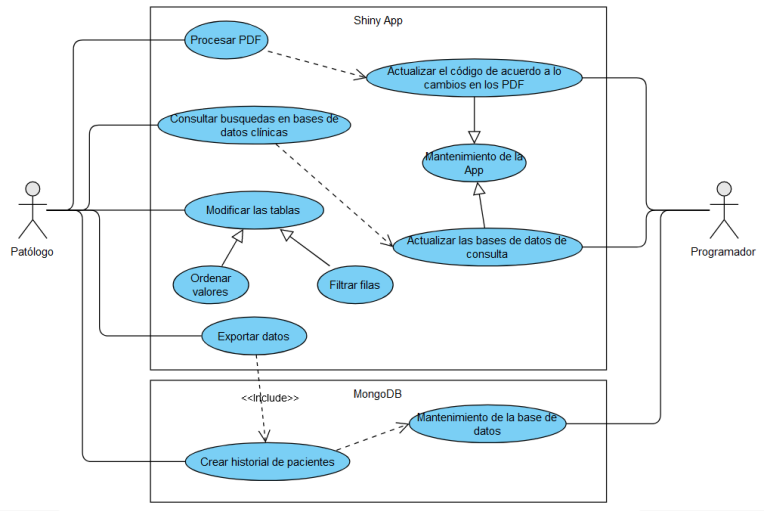


Figura E.1: Diagrama de casos de uso

E.2. Explicación casos de uso.

Patólogo

CU-01	**Procesar PDF**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-08
Descripción	Procesar PDF es la principal funcionalidad. Consiste en la detección y extracción de las variables requeridas y su posterior visualización dentro de la interfaz.
Precondición	Se deberá haber seleccionado algún documento PDF procedente del software Oncomine Reporter.
Acciones	1. Ejecutar la aplicación. 2. Pulsar la tecla espacio para eliminar la pantalla de inicio. 3. Presionar el botón "*Upload...*". 4. Seleccionar el/los archivo/s deseados de nuestro ordenador. 5. Oprimir la tecla enter. 6. Pulsar el botón "Analizar".
Postcondición	Esperar hasta que se muestren las tablas en pantalla.
Excepciones	Si la conexión a internet no es suficientemente buena, podría cerrarse la app al no poderse conectar con ClinVar.
Importancia	Alta

Tabla E.1: CU-01 Procesar PDF.

CU-02	**Modificar las tablas**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-01
Descripción	Una vez se obtienen las tablas, cada campo puede ser sobreescrito.
Precondición	Se debe haber analizado algún documento previamente.
Acciones	1. Pulsar dos veces sobre el dato que se desee sobreescribir. 2. Escribir lo que deseemos en el campo seleccionado y presionar enter.
Postcondición	Toda manipulación de las tablas que se haga después de la modificación tendrá en cuenta el nuevo valor.
Excepciones	Si pulsamos la tecla "ESC" o pulsamos cualquier lugar externo al campo de edición, el valor se almacena con los caracteres que tenga en ese momento escrito. Si queremos recuperar el dato anterior deberemos escribirlo a mano, no existe función de deshacer.
Importancia	Media

Tabla E.2: CU-02 Modificar las tablas.

CU-03	**Ordenar valores**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-01, RF-02
Descripción	Una vez se obtienen las tablas, cada columna puede ser ordenada ascendente o descendentemente.
Precondición	Se debe haber analizado algún documento previamente.
Acciones	1. Seleccionar la columna que queremos ordenar. 2. Si se presiona la cabecera de la columna 1 vez se marcará el símbolo "Δ" y, por lo tanto, se ordenará ascendentemente. 3. Si se presiona la cabecera de la columna 2 veces se marcará el símbolo "∇" y sus valores se ordenarán descendentemente.
Postcondición	Toda manipulación de las tablas que se haga después del nuevo orden tendrá en cuenta esta organización.
Excepciones	Si queremos volver al orden inicial no es posible. No obstante, sí que se puede ordenar por otras columnas. Si se presiona durante demasiado tiempo una cabecera, se eliminarán los datos de las tablas y se deberán volver a analizar los PDF.
Importancia	Baja

Tabla E.3: CU-03 Ordenar valores.

CU-04	**Filtrar filas**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-01, RF-02
Descripción	Una vez se obtienen las tablas, cada fila puede ser filtradas por las que contengan un determinado valor en sus columnas.
Precondición	Se debe haber analizado algún documento previamente.
Acciones	1. Marcar el campo que acompaña al texto "*Search*:" 2. Escribir el texto que se quiera buscar en las filas.
Postcondición	Toda manipulación de las tablas que se haga después del nuevo orden tendrá en cuenta solo las filas que aparezcan en pantalla.
Excepciones	Si el valor buscado se encuentra en distintos campos en distintas filas aparecerán ambas filas. No se puede filtrar solo una determinada columna, sino que se buscará el valor en todos los campos disponibles.
Importancia	Media

Tabla E.4: CU-04 Filtrar filas.

CU-05	**Consultar búsquedas en bases de datos clínicas**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-01, RF-09
Descripción	Al analizar los documentos seleccionados se buscarán las mutaciones de forma automática en la base de datos del NCBI proporcionando información relevante al sobre la variante al patólogo.
Precondición	Deberá tenerse conexión a internet para el correcto funcionamiento de la API.
Acciones	1. Seleccionar los archivos a procesar. 2. Presionar el botón "Analizar".
Postcondición	Los datos obtenidos de ClinVar se mostrarán en la columna "Patogenicidad.Buscada".
Excepciones	Si la conexión a internet no es suficientemente buena, podría cerrarse la app al no poderse conectar con ClinVar.
Importancia	Alta

Tabla E.5: CU-05 Consultar búsquedas en bases de datos clínicas.

CU-06	**Exportar datos**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-01, RF-02, RF-03, RF-04
Descripción	Las tablas son descargables en Excel o CSV. También existe la opción de impresión y de copiar en el portapapeles.
Precondición	Deberá seleccionarse la tabla que se quiere manipular.
Acciones	1. Posicionarse en la ventana de la tabla deseada. 2. Pulsar el botón "*Download*" para descargar la tabla, "*Copy*" para copiarla en el portapapeles o "*Print*" para imprimirla. 3. Seleccionar el formato de descarga. 4. Elegir el directorio de descarga. 5. Redactar el nombre del archivo. 6. Marcar el botón de guardar. 7.
Postcondición	Los datos serán almacenados en el mismo orden que se muestran en pantalla.
Excepciones	Si no hay datos para exportar, se descargarán archivos vacíos.
Importancia	Alta

Tabla E.6: CU-06 Exportar datos.

CU-07	**Crear historial de pacientes**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-01, RF-02, RF-04, RF-10
Descripción	Los datos mostrados en la app pueden ser almacenados en una base de datos para mantener un historial de pacientes.
Precondición	Hay que tener instalado MongoDBCompass.
Acciones	1. Posicionarse en la ventana de la tabla a almacenar. 2. Presionar el botón "Almacenar Datos"
Postcondición	Para visualizar los registros hay que ejecutar el comando "mongod" en la consola del ordenador para iniciar la base de datos.
Excepciones	Si el campo NHC de alguno de los pacientes a almacenar coincide con algún registro ya existente, ese paciente no será guardado de nuevo.
Importancia	Alta

Tabla E.7: CU-07 Crear historial de pacientes.

Programador

CU-08	**Actualizar el código de acuerdo a los cambios en los PDF**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-01, RF-02, RF-04
Descripción	Si el modelo de los PDF es modificado o se necesitan extraer nuevas variables, puede ser necesario trabajar el código.
Precondición	Hay que tener ligeras nociones de R.
Acciones	1. Abrir el archivo llamado app.r 2. Cambiar las variables de búsqueda de datos a nuestro gusto. 3. Guardar el documento. 4. Reiniciar la aplicación.
Postcondición	Hay que reiniciar la aplicación y volver a cargar los documentos para apreciar los cambios realizados.
Excepciones	Si la nueva expresión regular no está bien, la aplicación no va a ser capaz de encontrar lo que buscamos.
Importancia	Alta

Tabla E.8: CU-08 Actualizar el código de acuerdo a los cambios en los PDF.

CU-09	**Actualizar las bases de datos de consulta**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-05
Descripción	Las bases de datos están en constante crecimiento, lo que conlleva que aparezcan nuevas bibliotecas que nos sean de interés.
Precondición	Se debe conseguir acceso a las API.
Acciones	1. Registrarse en la base de datos 2. Solicitar acceso a la API. 3. Obtener las credenciales de acceso. 4. Documentarse sobre la API. 5. Configurar las solicitudes.
Postcondición	Hay que reiniciar la aplicación y volver a cargar los documentos para apreciar los cambios realizados.
Excepciones	Si las solicitudes no están bien programadas, es posible que no se encuentre nada.
Importancia	Alta

Tabla E.9: CU-09 Actualizar las bases de datos de consulta.

CU-10	**Mantenimiento de la base de datos**
Versión	2.0
Autor	Samuel González Martín
Requisitos asociados	RF-07
Descripción	Un funcionamiento eficiente y una administración adecuada de usuarios y registros son esenciales para mantener un historial preciso de pacientes y prevenir accesos no autorizados..
Precondición	Se debe saber usar MongoDB.
Acciones	1. Conectarse a la base de datos. 2. Acceder a la vista de administración de usuarios. 3. Crear un nuevo usuario. 4. Configurar los detalles del usuario 5. Asignar roles. 6. Guardar el nuevo usuario.
Postcondición	Hay que verificar que el usuario ha sido correctamente añadido.
Excepciones	Si no se ha almacenado correctamente, habrá que repetir de nuevo el proceso.
Importancia	Alta

Tabla E.10: CU-10 Mantenimiento de la base de datos.

E.3. Prototipos de interfaz o interacción con el proyecto.

La interfaz de la aplicación está compuesta por dos pantallas. La primera es una pantalla de inicio que muestra una imagen, como se ilustra en E.2. La segunda pantalla E.3 contiene el resto de la funcionalidad de la aplicación. Esta interfaz se caracteriza por su diseño sencillo y su facilidad de uso.

Figura E.2: Pantalla de inicio

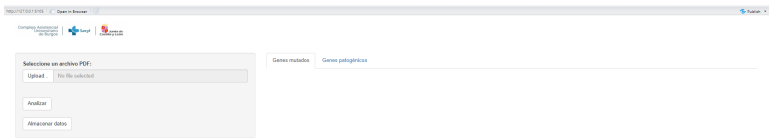


Figura E.3: Pantalla general

Apéndice F

Estudio experimental

F.1. Cuaderno de trabajo.

Una vez redactadas las técnicas empleadas en el desarrollo del software, cabe destacar que no todas ellas fueran la principal opción. Se han producido varias pruebas y errores en el proceso de creación, entre ellas cabe destacar:

- Al iniciar el proyecto hubo problemas a la hora de cargar y descargar archivos de la aplicación, dado que al ejecutarse en local, Shiny proporciona para estas acciones ubicaciones temporales para los archivos. Para solventar esta situación se encontró la biblioteca ShinyFiles que es una extensión de la propia aplicación de Shiny que proporciona una API para el acceso del cliente al sistema de archivos del servidor [thomasp85, 2022]. ShinyFiles no funcionó como se esperaba, por lo que fue sustituido por el paquete DT que es más intuitivo y permite descargar tablas más fácilmente.

- Durante todo este tiempo el departamento de anatomía patológica ha estado intentando implementar en conjunto con el resto de España un mismo modelo de PDF para los informes. Para evitar que se necesitaran modificar variables cada vez que un nuevo modelo era configurado en el software Oncomine Reporter se investigaron otras herramientas para la extracción de datos de estos documentos. Se probaron técnicas de reconocimiento avanzado de caracteres y de extracción de tablas de PDF, en específico se probaron los paquetes Tesseract y Tabulizer. Se testearon estos paquetes en un proyecto alterno de otra aplicación Shiny para una *hackathon* la cual acabo siendo ganada. Sin embargo,

los paquetes no reconocen a la perfección los elementos que queremos extraer y tienen un ratio demasiado alto de error, por lo que se descartó su uso para este TFG.

- Respecto al almacenamiento de la información de los pacientes en una base de datos, se comenzó trabajando con SQLite, pero al trabajar con datos no relacionales (*arrays* como atributos) ocurrían problemas de compatibilidad. Para solucionar este problema se plantearon dos estrategias, de las cuales se acabó optando por la segunda opción y pasando así de trabajar con SQLite a MongoDB:

  - Convertir los *dataframes* en datos relacionales, pasando cada atributo con *arrays* a un *dataframe* individual para ese atributo y relacionarlo con el *dataframe* principal mediante el número de historia clínica.
  - Trabajar con una base de datos noSQL que nos permita trabajar con datos no relacionales.

- Se propuso la idea de implementar API para la consulta de genes mutados con el objetivo de identificar aquellos de patogenicidad desconocida. Para ello, se consultó con el equipo de anatomía patológica sobre los recursos genómicos que más utilizaban, obteniendo como respuesta OncoKB, Varsome, Cbioportal, ClinVar y Franklin. De estas cuatro opciones, ClinVar fue la única que no presentó restricciones para el acceso a su API, por lo que ha sido la única que se ha podido integrar a la aplicación.

- Tras finalizar la programación de la visualización en pantalla de los *dataframes*, se observó que el rendimiento del código no era el esperado. Para solucionarlo, se optimizó mediante la aplicación de funciones de la familia apply, lo que permitió reducir significativamente el número de bucles. Gracias a este cambio, no solo se redujo el número de líneas de código, sino que también se logró disminuir el tiempo de espera a prácticamente la mitad.

F.2. Configuración y parametrización de las técnicas.

En este trabajo se llevan a cabo tres parametrizaciones fundamentales:

- La base de la aplicación se basa en la búsqueda y configuración de expresiones regulares repetidas en los informes que nos permitan extraer los datos requeridos. Estas expresiones regulares son asignadas a variables tipo *String*.
- La búsqueda en la base de datos requiere de una correcta configuración de búsqueda, bien mediante el nombre del gen, la codificación o el cambio de aminoácidos. La consulta en ClinVar se realiza con una cadena de texto que contenga el nombre del gen y una de las otras dos características.
- La conexión con la base de datos de MongoDB requiere indicar al programa cuál es la ruta a dicha base de datos. Esta ruta es proporcionada por MongoDB, pero en caso de necesitarlo se puede modificar y por consiguiente, también se debe cambiar en la app.

F.3. Detalle de resultados.

Todos los resultados ha sido expuestos en el apartado de resultados de la memoria.

Apéndice G

Anexo de sostenibilización curricular

G.1. Introducción

La realización de este proyecto me ha proporcionado una valiosa oportunidad para profundizar en mis habilidades y conocimientos en varios ámbitos tecnológicos, especialmente en el lenguaje de programación R. Aunque inicialmente no tenía una amplia experiencia en este lenguaje, su uso intensivo durante el desarrollo del proyecto ha sido instrumental para mi crecimiento profesional y académico. Ahora, tengo una mayor competencia en R, lo cual será fundamental para mis estudios de posgrado y futuras investigaciones.

Uno de los aspectos más destacados del proyecto ha sido el aprendizaje y la aplicación práctica de tecnologías de bases de datos noSQL. A diferencia de las bases de datos relacionales, las noSQL están diseñadas para manejar grandes volúmenes de datos no estructurados, proporcionando flexibilidad y escalabilidad. Este conocimiento me ha permitido comprender mejor cómo manejar y analizar datos no relacionales, una habilidad que es cada vez más relevante en la era del Big Data. La experiencia adquirida en la manipulación de estas bases de datos no solo ha ampliado mi perspectiva sobre la gestión de datos, sino que también me ha equipado con herramientas prácticas que podré aplicar en proyectos futuros.

Otra área en la que he adquirido conocimientos significativos es el uso de API (Interfaces de Programación de Aplicaciones). Antes de este proyecto, tenía un interés teórico en las API, pero no había tenido la oportunidad de trabajar con ellas de manera práctica. A lo largo del proyecto, he aprendido

a interactuar con ellas, comprendiendo sus estructuras y cómo integrarlas efectivamente en aplicaciones. Este aprendizaje es particularmente valioso, ya que las API son fundamentales para la interoperabilidad entre sistemas y la creación de aplicaciones robustas y escalables.

Además, durante el desarrollo de este proyecto, he tenido la oportunidad de aprender a desarrollar Shiny Apps en R. Shiny es un paquete de R que permite la creación de aplicaciones web interactivas directamente desde el lenguaje de programación R. La habilidad para desarrollar Shiny Apps es especialmente útil para visualizar y compartir resultados de análisis de datos de una manera intuitiva y accesible. El aprendizaje de esta tecnología no solo ha mejorado mi capacidad para comunicar hallazgos de datos, sino que también me ha proporcionado una herramienta poderosa para la creación de aplicaciones interactivas y visualizaciones dinámicas. Este conocimiento será de gran valor en mis estudios avanzados y en mi carrera profesional, permitiéndome presentar datos y análisis de una manera más impactante y efectiva.

La integración de estos conocimientos ha sido un proceso enriquecedor que me ha permitido ver la tecnología desde una perspectiva más amplia y aplicada. La combinación de habilidades en R, bases de datos noSQL, API, y desarrollo de Shiny Apps me posiciona de manera ventajosa para abordar desafíos complejos en mis futuros estudios de maestría y en el ámbito profesional.

Más allá de los conocimientos técnicos, este proyecto ha reforzado mi capacidad de autogestión y resolución de problemas. La necesidad de investigar, aprender y aplicar nuevos conceptos de manera autónoma ha sido una experiencia valiosa que me ha preparado mejor para los rigores de los estudios avanzados. La experiencia de trabajar en un proyecto tan completo me ha dado una mayor confianza en mi capacidad para enfrentar y superar desafíos académicos y profesionales.

Bibliografía

[APD, 2024] APD, R. (2024). ¿qué es la metodología scrum y cómo aplicarla? | apd. https://www.apd.es/metodologia-scrum-que-es/. Acceso realizado el 29 de mayo de 2024.

[CRAN, 2024] CRAN (2024). Download r-4.4.0 for windows. the r-project for statistical computing. https://cran.r-project.org/bin/windows/base/. Acceso realizado el 5 de junio de 2024.

[GitHub, 2024] GitHub (2024). Clonar un repositorio - documentación de github. https://docs.github.com/es/repositories/creating-and-managing-repositories/cloning-a-repository. Acceso realizado el 11 de junio de 2024.

[MongoDB, 2024] MongoDB (2024). Download mongodb atlas cli | mongodb. https://www.mongodb.com/try/download/atlascli. Acceso realizado el 5 de junio de 2024.

[Posit, 2024] Posit (2024). Rstudio desktop - posit. https://posit.co/download/rstudio-desktop/. Acceso realizado el 5 de junio de 2024.

[Richardson et al., 2024] Richardson, N., Cook, I., Crane, N., Dunnington, D., François, R., Keane, J., Moldovan-Grünfeld, D., Ooms, J., Wujciak-Jens, J., and Apache Arrow (2024). *arrow: Integration to 'Apache' 'Arrow'*. R package version 16.1.0, https://arrow.apache.org/docs/r/.

[Rstudio, 2023] Rstudio (2023). Scales coordinate systems. Acceso realizado el 23 de mayo de 2024.

[thomasp85, 2022] thomasp85 (2022). thomasp85/shinyfiles: A shiny extension for server side file access. https://github.com/thomasp85/shinyFiles. Acceso realizado el 11 de junio de 2024.

[Wickham, 2020] Wickham, H. (2020). Welcome | mastering shiny. https://mastering-shiny.org/index.html. Acceso realizado el 23 de mayo de 2024.

[Wickham et al., 2023] Wickham, H., Girlich, M., Fairbanks, M., and Dickerson, R. (2023). *dtplyr: Data Table Back-End for 'dplyr'*. https://dtplyr.tidyverse.org, https://github.com/tidyverse/dtplyr.

[Wickham and Henry, 2023] Wickham, H. and Henry, L. (2023). *purrr: Functional Programming Tools*. R package version 1.0.2, https://github.com/tidyverse/purrr.