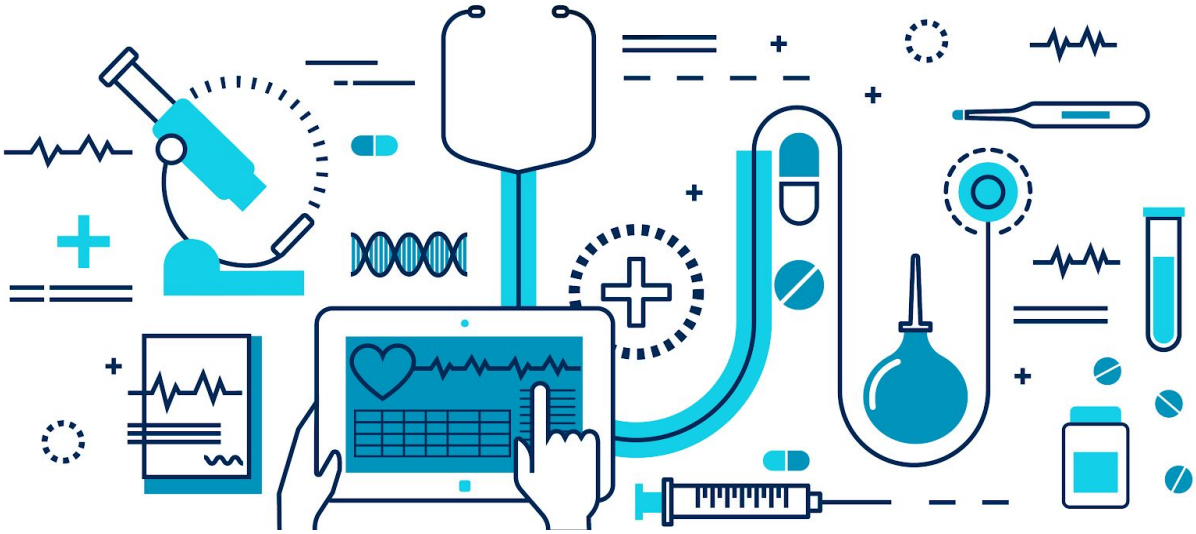
Memoria

M2.974 - TFM Estudios clínicos

Ciencia de datos aplicada a la Salud

PAC4



Autor:
Hector Heranz Cayuela

Directora del proyecto:
Susana Pérez Álvarez

Master de Ciencia de Datos (Data Science)
UOC
Barcelona, Junio de 2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Inteligencia de estudios clínicos*
Nombre del autor:	*Hector Herranz Cayuela*
Nombre del consultor/a:	Susana Pérez Álvarez
Nombre del PRA:	Àngels Rius Gavidia
Fecha de entrega (mm/aaaa):	06/2019
Titulación::	Máster universitario en Ciencia de Datos
Área del Trabajo Final:	*Ciencia de datos aplicada a la Salud*
Idioma del trabajo:	Español
Palabras clave	*Estudios clínicos, ciencia de datos, inteligencia*

Resumen del Trabajo (máximo 250 palabras):

Los estudios clínicos o ensayos clínicos son necesarios para evaluar la eficacia y seguridad de un tratamiento. Sin embargo, a pesar de la existencia de portales de registros clínicos, estos estudios tienen dificultades en el reclutamiento de voluntarios, en la utilidad de los datos y en la visión general de los datos. En este proyecto, se ha realizado una solución denominada Inteligencia de Estudios Clínicos (Clinical Trials Intelligence), que consiste en una plataforma web para la inteligencia de estudios clínicos, donde se ofrecen los datos y el conocimiento extraído del conjunto de datos, respondiendo a diferentes temas de interés para los involucrados. Para lograr este objetivo, se ha implementado una ETL sobre los datos de ensayos estudios ofrecidos por un portal de registros clínicos que opera a nivel mundial. Los datos se extraen, se procesan y se transforman de forma periódica para generar el conocimiento que se muestra a través de visualizaciones interactivas en la plataforma web, permitiendo las partes interesadas resolver cuestiones clave que pueden ayudar a la prosperidad de los estudios y la difusión de la información.

Abstract (in English, 250 words or less):

Clinical studies or clinical trials are necessary to evaluate the efficacy and safety of a treatment. However, despite the existence of portals of clinical records, these studies have difficulties in the recruitment of volunteers, in the usefulness of the data and in the general overview of the data. In this project, a solution called Clinical Trials Intelligence was developed, which consists of a web platform for the intelligence of clinical studies, where the data and knowledge extracted from the data set are offered, responding to different topics of interest to those involved. To achieve this goal, an ETL has been implemented on the data of trials studies offered by a portal of clinical records that operates worldwide. The data is extracted, processed and transformed periodically to generate the knowledge that is shown through interactive visualizations on the web platform, allowing interested parties to resolve key issues that can help the prosperity of studies and dissemination of information.

Índice

1. Introducción

1.1 Interés y relevancia

El número de estudios clínicos está creciendo en los últimos años, cada vez con mayor intensidad [1]. Los estudios clínicos mueven millones de dólares anualmente, y son de vital importància para la investigación en medicina [2]. A pesar de que existen datos abiertos sobre registros clínicos al menos desde 2008 [3], no se encuentran aplicativos de datos abiertos de cara al público general más allá de los propios organismos que mantienen las bases de datos permitiendo utilidades de consulta básica sobre los datos, y las empresas que se dedican a vender los datos o estudios concretos sobre los datos como servicio.
En esta situación, es posible ofrecer un sistema que aporte una mejor visión sobre el conocimiento que se genera sobre los estudios clínicos, con el fin de cubrir los intereses de diferentes colectivos de usuarios y facilitar el acceso a la información, tanto a profesionales como pacientes o interesados en el área.

1.2 Alcance y objetivos

Este proyecto se centra en el desarrollo de una *Proof Of Concept (POC)*, en la que utilizando la versión de acceso libre de elementos en la nube de forma modular, se implementa un sistema conceptualmente estable y robusto en el que se responden preguntas sobre los estudios clínicos en formato gráfico a través de la web. No se tiene en cuenta la ineficiencia de las partes utilizadas en su versión gratuita, ya que se pueden escalar con versiones de pago y de esta forma mejorar tiempos de respuesta web, seguridad, etc. Tampoco se tiene en cuenta matices o acabados de usabilidad finales. En cambio se valora la estratégia de uso de los componentes que forman el sistema y sus posibilidades de escalado y compatibilidad, de la misma forma que el modelo conceptual de la presentación y las visualizaciones. Es decir, se valora la construcción de la base del sistema.
Por lo tanto, el proyecto se limita a la investigación sobre las posibles fuentes de datos, la generación de un proceso *ETL* viable, y la visualización e interacción con respuestas a preguntas sobre los estudios clínicos. Con el objetivo de internacionalizar la presentación de los datos, ésta se realiza en inglés.

2. Metodología y planificación

Para el desarrollo de este proyecto se basa en scrum de 1 persona [4], en el que para la parte de desarrollo, en cada sprint se realiza una iteración sobre un modelo de prototipado [5], siendo el desarrollador su propio usuario. Es una metodología del desarrollo del proyecto muy flexible pero ordenada para adaptarse fuera de un horario laboral, a la vez que se va descubriendo el proyecto durante su realización.
Esta metodología ha sido aplicada sobre la planificación del proyecto, que se basa en la entrega de las diferentes partes del proyecto, mostrado en la *Figura 1*.

Figura 1: Planificación de entregas del proyecto

El proyecto se componte de las siguientes 5 etapas, en las que se requiere la entrega de la misma:

PAC1 - Definición y planificación del proyecto final

En esta etapa se planifica el proyecto gestionando los recursos para poder cumplir con los objetivos principales en la fecha propuesta.

A modo de guía, se plantean unas preguntas iniciales que el proyecto deberá resolver. Estas preguntas podrán evolucionar y se podrán añadir de nuevas a lo largo del proyecto, una vez se tenga más conocimientos sobre el contexto.

Los objetivos secundarios se realizarán en caso de suficiente excedente de recursos para realización de los mismos.

Periodo: 20/02/2019→ 03/03/2019

PAC2- Estado del arte

En esta etapa se buscan y analizan infografías, sitios web, o cualquier recurso similar que pueda contestar a las preguntas planteadas o sugiera de nuevas.

Adicionalmente y relacionado con el tipo de búsqueda, se evalúan posibles candidatos a fuentes de datos y la forma de obtención de los mismos, además de comprobar los datos concretos que se ofrecen.

Periodo: 04/03/2019→ 24/03/2019

PAC3- Diseño e implementación del trabajo

Esta es la etapa con más peso del proyecto.

1. **Análisis**. En primer lugar y después de tener el conocimiento de la etapa anterior, se decidirán qué herramientas son las idóneas para resolver los objetivos principales del proyecto, y ver que opciones serían viables para desarrollar. Este paso ha de ocupar alrededor de un 5% - 10% del tiempo. (3-6 días)
2. **Diseño**. En segundo lugar estaría el diseño del sistema por completo, con una precisión ajustada al nivel de conocimiento del que se disponga. Este paso ha de ocupar entre un 15-20% del tiempo.(9-12 días)
3. **Programación**. En tercer lugar se encontraría la implementación del sistema. Se dedica entre un 40% y 50% del tiempo, desarrollando por iteraciones como en una metodología de prototipado, incluyendo ciertas pruebas. (24-30 días)
4. **Pruebas**. En este cuarto paso haría las pruebas finales, dedicando un 15% del tiempo a finalizar el desarrollo. (9 días)
5. **Sobrecarga**. Otro 15% del tiempo sería guardado como margen para imprevistos, pequeñas mejoras o arreglos. (9 días)

Periodo: 25/03/2019 → 19/05/2019

PAC4- Redacción de la memoria

En esta etapa se redacta una memoria del proyecto, y se prepara la presentación y defensa del mismo.

Periodo: 20/05/2019→ 09/06/2019

PAC5- Presentación y defensa del proyecto

En esta etapa se desarrolla la presentación y defensa el proyecto.

Periodo: 10/06/219→ 16/06/2019

3. Estado del arte

En este apartado se realiza un repaso del estado del arte en cuanto a los portales de estudios clínicos así como de los temas directamente relacionados con los mismos. El objetivo de este apartado es presentar una fotografía del estado actual sobre el tema. De este modo se permite utilizar como fuente de datos uno o varios de estos portales de estudios clínicos en base a la compatibilidad con el proyecto, además de conocer otros trabajos similares al que se pretende realizar.

3.1 Categorización de los portales de registros clínicos.

El "**International Committee of Medical Journal Editors**" (ICMJE) es un pequeño grupo de editores de revistas médicas generales y representantes de organizaciones relacionadas seleccionadas que trabajan en conjunto para mejorar la calidad de la ciencia médica y sus informes.

En las siguientes tablas, los registros reconocidos por el "**International Committee of Medical Journal Editors**" se encuentran coloreados en azul.

Nivel internacional

Agencia	**Portal de búsqueda**
International Committee of Medical Journal Editors http://icmje.org	No dispone de portal de búsqueda propio.
World Health Organization (WHO) – International Clinical Trials Registry Platform: http://www.who.int/ictrp/en/	http://apps.who.int/trialsearch/
World Medical Association http://www.wma.net	No dispone de portal de búsqueda propio.
Global Clinical Trials (GCT) Data https://www.globalclinicaltrialsdata.com/	https://globalclinicaltrialdata.com/

América del Norte

Región / País	Nombre del registro
United States	ClinicalTrials.gov: https://www.clinicaltrials.gov/ct2/home
Canada	Health Canada Clinical Trial Database: http://www.hc-sc.gc.ca/dhp-mps/prodpharma/databasdonclin/index-eng.php

Unión Europea

Región / País	Nombre del registro
European Union	EU Clinical Trials Register: https://www.clinicaltrialsregister.eu/
Germany	German Clinical Trials Register: https://drks-neu.uniklinik-freiburg.de/drks_web/
Netherlands	Netherlands Trial Register (Dutch): http://www.trialregister.nl/trialreg/index.asp
Switzerland	Swiss National Clinical Trials Portal: http://www.kofam.ch/en/swiss-clinical-trials-portal.html
United Kingdom	ISRCTN: http://www.isrctn.com/
Spain	Registro Español de estudios clínicos (REec) https://reec.aemps.es

Asia / Pacífico / Medio Oriente

Región / País	Nombre del registro
Australia	Australian New Zealand Clinical Trials Registry: http://www.anzctr.org.au/
China	Chinese Clinical Trial Registry: http://www.chictr.org.cn/enIndex.aspx
India	Clinical Trials Registry – India: http://ctri.nic.in/
Iran	Iranian Registry of Clinical Trials: http://www.irct.ir/
Japan	Japan Primary Registries Network: http://rctportal.niph. go.jp/
Korea	Clinical Research Information Service: https://cris.nih.go.kr/cris/en/use_guide/cris_introduce.jsp
New Zealand	Australian New Zealand Clinical Trials Registry: http://www.anzctr.org.au/
Philippines	Philippine Health Research Registry: http://registry.healthresearch.ph/
Sri Lanka	Sri Lanka Clinical Trials Registry: http://www.slctr.lk/

Thailand	Thai Clinical Trials Registry: http://www.clinical trials.in.th/

América Latina / Caribe

Región / País	Nombre del registro
Brazil	Brazilian Clinical Trials Registry: http://www.ensaiosclinicos.gov.br/
Cuba	Public Cuban Registry of Clinical Trials: http://registroclinico.sld.cu/en/home
Peru	Peruvian Registry of Clinical Trials: https://www.ins.gob.pe/ensayosclinicos/

Africa

Región / País	Nombre del registro
Pan Africa	Pan African Clinical Trials Registry: http://www.pactr.org/
South Africa	South African National Clinical Trials Register: http://www.sanctr.gov.za/
Tanzania	Tanzania Clinical Trial Registry: http://www.tzctr.or.tz/

Otros relacionados

Región / País	Nombre del registro
Spain - Madrid	Asociación Madrileña de Hematología y Hemoterapia (AMHH) Ensayos Clínicos Activos (ECA) https://www.ensayoshematologiamadrid.es
Spain	Sistema de información de ensayos Clínicos de Hematología y Hemoterapia. (Reec + EudraCT) https://www.hemotrial.es
Europe	EudraCT (European Union Drug Regulation Authorities Clinical Trials) https://eudract.ema.europa.eu

Existen portales de registros clínicos orientados a diferentes granularidades y/o zonas geográficas. Por otro lado también existen portales orientados a una enfermedad o tipo de enfermedades en concreto, los cuales no se profundizan en este proyecto.
Dentro del àmbito geográfico y de las agencias, se pueden encontrar ciertos portales "origen" de los que dependen otros tipos de portales que incorporan estos datos, y ciertos portales de "red de registros" que hacen uso de múltiples portales "origen".
En un nivel más profundo existen "Protocolos", de los que no se entra en materia en el desarrollo de este proyecto.

El **International Committee of Medical Journal Editors** es una entidad que otorga reconocimiento y validez a los portales de registros clínicos que cumplen ciertos requisitos, siendo los portales reconocidos por esta entidad los más relevantes.

Los portales más reconocidos a nivel continental y de nuestro interés son https://www.clinicaltrials.gov de Estados Unidos y https://www.clinicaltrialsregister.eu/ a nivel Europeo. Ambos son válidos para utilizar como origen de datos, ya que recogen gran cantidad de registros. A priori, destaca *clinicaltrials.gov* ya que es un proveedor de datos para la *International Clinical Trials Registry Platform* (*ICTRP*) de la *WHO.*

Por otro lado, el uso de portales a nivel internacional parece la mejor opción, ya que recogen multitud de registros a nivel internacional. Estos portales internacionales principalmente son *WHO* y *GCT*.

Cabe destacar que de optar por realizar un proyecto exclusivamente orientado a nivel nacional, se podría optar por los datos de los registros Reec o la plataforma ECA-AMHH.

3.2 Evaluación de los portales de registros clínicos.

En este proyecto se busca la mayor cobertura y calidad posibles. Por ello, éste se orienta hacia portales con cobertura internacional que ofrezcan la mayor calidad posible. En los siguientes apartados se tienen en cuenta los portales que cumplen estos requisitos

Nivel internacional

1. World Health Organization (WHO)

La red de registros de ***WHO*** está compuesta por:

- Registros primarios (Reconocidos por ICMJE)
    - Australian New Zealand Clinical Trials Registry (ANZCTR)
    - Brazilian Clinical Trials Registry (ReBec)
    - Chinese Clinical Trial Registry (ChiCTR)
    - Clinical Research Information Service (CRiS), Republic of Korea
    - Clinical Trials Registry - India (CTRI)
    - Cuban Public Registry of Clinical Trials(RPCEC)
    - EU Clinical Trials Register (EU-CTR)
    - German Clinical Trials Register (DRKS)
    - Iranian Registry of Clinical Trials (IRCT)
    - ISRCTN
    - Japan Primary Registries Network (JPRN)
    - Thai Clinical Trials Registry (TCTR)
    - The Netherlands National Trial Register (NTR)
    - Pan African Clinical Trial Registry (PACTR)
    - Peruvian Clinical Trial Registry (REPEC)
    - Sri Lanka Clinical Trials Registry (SLCTR)

- Registros de socios
    - Centre for Clinical Trials, Clinical Trials Registry - Chinese. University of Hong Kong. Affiliated registry: ChiCTR
    - The Acupuncture-Moxibustion Clinical Trial Registry (AMCTR). Beijing. Affiliated registry: ChiCTR

- Proveedores de datos
    - Australian New Zealand Clinical Trials Registry (ANZCTR)
    - Brazilian Clinical Trials Registry (ReBec)
    - Chinese Clinical Trial Register (ChiCTR)
    - Clinical Research Information Service (CRiS), Republic of Korea
    - ClinicalTrials.gov (Reconocido por ICMJE)
    - Clinical Trials Registry - India (CTRI)
    - Cuban Public Registry of Clinical Trials (RPCEC)
    - EU Clinical Trials Register (EU-CTR, Reconocido por ICMJE))
    - German Clinical Trials Register (DRKS)
    - Iranian Registry of Clinical Trials (IRCT)
    - ISRCTN
    - Japan Primary Registries Network (JPRN)
    - Pan African Clinical Trial Registry (PACTR)
    - Peruvian Clinical Trials Registry (REPEC)
    - Sri Lanka Clinical Trials Registry (SLCTR)
    - Thai Clinical Trials Register (TCTR)
    - The Netherlands National Trial Register (NTR)

- Registros que trabajan con el *ICTRP* para convertirse en registros primarios

WHO dispone de una cobertura geográfica excelente. Explora ~472.000 estudios clínicos.

Sobre la calidad y accesibilidad del registro se destaca:

- ***WHO*** dispone de un aplicativo web para realizar búsquedas sobre los registros: http://apps.who.int/trialsearch/
- Hacer *crawling* sobre la base de datos *ICTRP* ahora requiere un nombre de usuario / contraseña. Para solicitar acceso a crawlear páginas, se debe enviar un correo electrónico a ictrpinfo@who.int. (Ver apéndice 1)
- Los datos se actualizan con una frecuencia que puede variar entre 1 a 4 semanas, dependiendo de la fuente de datos.
- Búsquedas exportables en CSV y XML. Términos y condiciones prohíben uso comercial.
- 21 Atributos sobre cada ensayo clínico y acceso a la fuente original.
- El directorio de crawling es un listado de enlaces
- El acceso comercial al crawling tiene un coste de 1000$ anuales

2. Global Clinical Trials (GCT)

La red de registros de ***GCT*** está compuesta por:

- www.anzctr.org.au
- www.chictr.org.cn
- www.chinadrugtrials.org.cn
- www.clinicaltrialsregister.eu
- www.clinicaltrials.gov
- www.clinicaltrials.jp

GTC dispone de una red de registros que es bastante limitada en comparación con *WHO*, pero dispone de acceso a las dos fuentes que hemos concluido que son más relevantes para nosotros, *clinicaltrialsregister.eu* y *clinicaltrials.gov* . Explora ~300,000 estudios clínicos.

Sobre la calidad y accesibilidad del registro se destaca:

- ***GCT*** dispone de un aplicativo web para realizar búsquedas sobre los registros: https://globalclinicaltrialdata.com
- En la descarga de datos no se permite uso comercial sin previa autorización.
- La descarga de datos no funciona (13/03/2019)
- No menciona tiempos de actualización, pero en los resultados de búsqueda aparecen registros con fecha del mismo dia.
- 29 Atributos visibles sobre cada estudio clínico. No hay acceso directo a la fuente original.

América del Norte

1. ClinicalTrials.gov

La base de datos abarca registros a nivel mundial con la siguiente distribución (30/5/2019):

Nombre de la región	Numero de estudios
Mundo	**307,060**
África	8.550
Centroamérica	2,867
este de Asia	33,559
Japón	5,469
Europa	87,222
medio este	12,648
Norteamérica	135,133
Canadá	20,190
Groenlandia	1
Mexico	3,355
Estados Unidos	121,772
Norte de asia	5,366
Pacifica	7,300
Sudamerica	9,922
Asia del Sur	4,653
El sudeste de Asia	6,126

Explora más de 300.000 estudios de investigación en los 50 estados y en 210 países.

Sobre la calidad y accesibilidad del registro se destaca:

- ***ClinicalTrials.gov*** dispone de un aplicativo web para realizar búsquedas sobre los registros:https://www.clinicaltrials.gov/ct2/home
- En los términos y condiciones no especifica condiciones sobre el uso comercial de forma directa.
- Permite acceder a los datos mediante *API* o descargar todos los datos públicos en un fichero comprimido de aprox. 1.2 GB.
- 45 atributos visibles sobre cada estudio clínico en la visualización tabular.

Unión Europea

1. EU Clinical Trials Register

La base de datos tiene un alcance a nivel europeo, disponiendo de:
~35.00 estudios clínicos que se encuentran en la base de datos de EudraCT.
~19.000 estudios clínicos realizados fuera de la UE / EEE que están vinculados al desarrollo europeo de medicina pediátrica.

Sobre la calidad y accesibilidad del registro se destaca:

- ***EU Clinical Trials Register*** dispone de un aplicativo web para realizar búsquedas sobre los registros: https://www.clinicaltrialsregister.eu/ctr-search/search
- Los términos y condiciones hacen referencia a *"European Medicines Agency"*, y este permite el uso comercial.
- Permite un acceso similar a *API* y una descarga de ficheros poco compatible para el procesado de datos (Texto plano, desestructurado).
- 150-200 atributos visibles sobre cada estudio clínico, en un formato de texto plano.

De los registros seleccionados, llegamos a las siguientes conclusiones:

World Health Organization (WHO) es una organización a nivel internacional y acopla portales de numerosos países. No obstante, la actualización de datos parece lenta, sólo disponen de 21 atributos por estudio clínico y el acceso a los datos se ve algo limitado. El directorio de crawling es un buen recurso para indexar la información del portal, pero no para el análisis de datos ya que se requiere la descarga de los mismos.

Global Clinical Trials (GCT) es similar a *WHO* pero con menos cobertura y en las pruebas realizadas la descarga de datos producía un error.

ClinicalTrials.gov dispone de una cobertura similar a los portales internacionales. Cuenta con 45 atributos disponibles por estudio clínico, además de todo tipo de facilidades para el acceso y descarga de los datos.

EU Clinical Trials Register dispone de una cantidad de registros significativamente inferior a los portales anteriormente mencionados, un exceso de atributos por estudio clínico y una accesibilidad a los datos poco adecuada para la automatización (Datos en texto plano, no estructurados).

Por lo tanto, ***ClinicalTrials.gov*** es el mejor candidato a fuente de datos para el proyecto.

3.3 Análisis de ClinicalTrials.gov

Tal y como se ha indicado, *ClinicalTrials.gov* es el mejor candidado a fuente de datos. Dependiendo de las funcionalidades, datos, formatos de datos y otros recursos que este ofrezca, será más o menos complejo desarrollar una arquitectura u otra para dar respuesta a unas u otras preguntas. A continuación se analiza el portal con el objetivo de sintonizar el proyecto con el mismo.

3.3.1 Cuestiones a responder

Al elegir *ClinicalTrials.gov* como fuente, disponemos de una información concreta sobre los estudios clínicos la cual permite tanto descubrir cómo resolver cuestiones sobre los mismos. La estructura de datos puede encontrarse en:
https://clinicaltrials.gov/ct2/html/images/info/public.xsd
Este fichero ofrece un historial de cambios sobre la estructura de datos, seguida por la estructura de los diferentes atributos que se utilizan.

En la definición y planificación del proyecto, se plantearon diferentes puntos de vista desde los que responder preguntas. En primer lugar se requería una base de conocimiento donde poder alzar las preguntas, una vez conocidos los datos de los que se dispone. Se mencionaron los siguiente puntos de vista de los que partir: Voluntarios, curiosos y expertos. En ***ClinicalTrials.gov*** se suguieren los siguientes: Para pacientes y familiares, Para investigadores y Para administradores de registros de estudio. Tras replantear, los puntos de vista finales son: **Pacientes y familiares, interesados/vistazo general, e investigadores y expertos**.

1. Desde el punto de vista de Pacientes y familiares se dará respuesta a preguntas orientadas a una enfermedad y su progreso según los estudios clínicos, así como el descubrimiento de qué investigadores se encuentran estudiándola. El objetivo es ayudar a entender la situación de la enfermedad.

2. Desde el punto de vista de interesados se dará respuesta a preguntas orientadas a una vista general sobre qué estudios se encuentran abiertos y sobre qué temas. El objetivo es promover o hacer visibles los estudios en curso y conocer cuales son los temas de estudio en la actualidad. Mostrar una vista general.
3. Desde el punto de vista de investigadores y expertos se dará respuesta a preguntas sobre cuáles son los principales patrocinadores e investigadores de un tema y en qué otros temas participan dichos patrocinadores e investigadores. El objetivo es ofrecer datos estratégicos al investigador.

Estas líneas pueden seguir evolucionando a lo largo del proyecto.

ClinicalTrials.gov **tiene un alcance internacional** muy amplio, por lo que tiene sentido plantear preguntas sobre datos o metadatos que generan el conjunto de estudios clínicos sobre el tiempo. Respecto a este punto, surgen cuestiones como:

- ¿Cómo ha evolucionado el número de estudios clínicos sobre [X enfermedad o condición] en los últimos años? Podemos generalizar la pregunta un nivel agrupando las enfermedades por categoría. Por ejemplo, las enfermedades minoritarias tienen una categoría en el buscador: (https://clinicaltrials.gov/ct2/search/browse?brwse=ord_alpha_all)

A continuación, se definen 3 preguntas concretas sobre cada punto de vista. Estas preguntas definen el concepto. De cara a la presentación de los datos, la formulación de las preguntas varía por la traducción al inglés y las decisiones finales. Si asociamos cada punto de vista a un *Dashboard*, cada pregunta resuelta se podría proyectar en una gráfica.

Pacientes y familiares.

1. ¿Cómo ha evolucionado el número de estudios clínicos sobre [X enfermedad o condición] en los últimos años?
2. ¿Cuántos estudios clínicos ha realizado cada investigador sobre [X enfermedad o condición] ?
3. ¿Dónde se están realizado o se han realizado estudios clínicos en los últimos [Y meses] sobre [X enfermedad o condición]?

Interesados / vistazo general (*overview*)

1. ¿Qué enfermedades o temas son los que más se están estudiando actualmente?
2. ¿Qué países tienen más estudios en curso actualmente?
3. ¿Cómo ha evolucionado el número de estudios clínicos en global?

Investigadores y expertos

1. ¿Qué entidades están patrocinando más estudios clínicos?
2. ¿Qué otros investigadores se encuentran trabajando sobre [X enfermedad o condición] o [Y fármaco] o [Z prueba]?

3. ¿En qué otros temas se encuentra trabajando o ha trabajado [X investigador] ? (Pendiente de acotar o subdividir)

En caso de disponer de suficientes recursos, adicionalmente se podrían aplicar técnicas de *Natural Language Processing (NLP)* para extraer palabras clave o resúmenes sobre enfermedades, temas que trata un investigador, etc. **Se añade entonces a los objetivos secundarios el ampliar las respuestas ofrecidas.**

3.3.2 Exploración técnica del buscador y de la API

ClinicalTrials.gov dispone de diferentes formas de acceder a los estudios:

1. Nueva búsqueda
    a. Se trata de una búsqueda simple por estado, condición, términos y país.
2. Búsqueda avanzada
    a. Se trata de una búsqueda por más de 25 atributos del estudio.
3. Estudios por tema
    a. Muestra un menú navegable de temas de estudio.
4. Estudios en el mapa
    a. Muestra un mapa de calor con el número total de estudios por territorio, accessible a diferente nivel de granularidad.

Adicionalmente, cuenta con un apartado de *"Trends, Charts, and Maps"* con una versión simple y principalmente enfocada a Estados Unidos de:

- Ubicaciones de los estudios registrados
- Ubicaciones de los estudios de reclutamiento
- Mapa de estudios registrados en *ClinicalTrials.gov*
- Tipos de estudios registrados
- Número de estudios registrados a lo largo del tiempo
- Número de estudios registrados con resultados publicados a lo largo del tiempo

ClinicalTrials.gov también cuenta con un apartado llamado *"Downloading Content for Analysis"* donde se explica el uso de la API.
Las opciones para la descarga de datos son:

1. Los 10 principales, 100, 1,000 o 10,000 estudios recuperados por su búsqueda.
2. Todos los estudios recuperados por su búsqueda (hasta un máximo de 10,000 registros de estudio)
3. Opciones para descargar contenidos de un estudio.(formatos pdf, texto, tsv, csv, xml)
4. Use los parámetros de URL para mostrar y guardar datos
5. Mostrar un solo registro en XML
6. Descargar múltiples registros en XML

7. Descargar todo el contenido del registro de estudio para su análisis (1-2Gb). Contiene múltiples directorios con ficheros en formato xml.

Se puede concluir que la accesibilidad a los datos de ***ClinicalTrials.gov*** permite flexibilidad a la hora de diseñar una arquitectura para el proyecto. No obstante, la descarga completa de los datos es la mejor opción cuando se incluyen preguntas sobre el histórico de estudios, y no se requiere tiempo real contra la fuente de datos.
Las propuestas que surgen entonces para encaminar el proyecto son las siguientes:

Propuesta 1 (La que se realiza):
La opción de acceso a los datos que más se adecua a los objetivos de esta propuesta es la descarga y procesado de todos los datos en forma de *ETL* programada diariamente, por ejemplo con *PDI Spoon* [6] o en Python.
Una vez descargados y procesados en local, es posibles publicarlos a *Google Sheets* [7] y utilizar *Tableau Public* [8] con *Google Sheets* para ofrecer visualizaciones con actualización diaria.
Estas visualizaciones se pueden incrustar en una web, por ejemplo una página web realizada con Wordpress con una plantilla con un estilo orientado como un *Dashboard* para que se encuentren agrupadas y presentadas al público.

Propuesta 2 (Alternativa):
Reenfocado hacia desarrolladores, crear un kernel en *Kaggle* [9] donde se incluya desde la *ETL* hasta la visualización de los datos, basado en las preguntas anteriormente realizadas.

Propuesta 3 (Alternativa):
Realizar el estudio en local, utilizando herramientas ad hoc y presentar una infografía sobre un tema en concreto, basado en las preguntas anteriormente realizadas.

3.4 Competidores / Otros recursos relacionados

Otros recursos de internet que ofrecen un resultado similar al que se busca conseguir con este proyecto son los siguientes:

3.4.1 Clarivate Analytics

https://clarivate.com/products/cortellis/cortellis-clinical-trials-intelligence/

El producto en concreto es **Cortellis Clinical Trials Intelligence**, que "*es un recurso poderoso para acelerar las decisiones de desarrollo clínico, informar la estrategia de cartera, seleccionar sitios de prueba y proporcionar inteligencia competitiva clave. La información de ensayos*

clínicos curada por expertos que cubre todas las indicaciones se integra con otra inteligencia científica y competitiva de Clarivate Analytics, para que pueda tomar las decisiones más informadas para maximizar sus inversiones clínicas y reducir el riesgo".

También trata cerca de 300.000 registros, la misma aproximación de número de estudios que trata este proyecto utilizando ***ClinicalTrials.gov*** como fuente de datos.

Visualmente se puede identificar que las visualizaciones están hechas con Tableau, que es una de las posibles propuestas que he hecho en cuanto al componente de visualización.

3.4.1 Otros similares

Existen productos similares a *Cortellis Clinical Trials Intelligence*, algunos de ellos listados a continuación:
http://e-mergeglobal.com/scientific-information/clinical-trials-intelligence/
https://pharmaintelligence.informa.com/products-and-services/data-and-analysis/trialtrove
http://longtaal.com/longtaal-provides-industrys-first-class-clinical-trials-intelligence/

Se puede concluir que, al parecer, el proyecto podría ofrecer de forma pública un recurso similar al que se encuentran vendiendo algunas empresas. La **Propuesta 1** tiene el potencial de cubrir algunas de las respuestas ofrecidas en los productos empresariales. Para ellos, posiblemente sea necesario abordar nuevos puntos de vista, como el del inversor/patrocinador. Por otra parte en internet también se encuentran infografías y otros recursos sobre temas concretos, que no serán abordadas en este proyecto.

4. Diseño e implementación del trabajo

En este apartado se resume el desarrollo del proyecto, incluyendo los objetivos y técnicas aplicadas. Así mismo, se hacen referencias a herramientas y se explican las problemáticas encontradas y cómo se han ido resolviendo.

Como metodología de trabajo se han seguido Iteraciones de prototipage según los sprints del scrum de 1 persona. En cada sprint se definen diferentes tareas de desarrollo sobre un objetivo. Por lo tanto se han pasado por las etapas de análisis, diseño, programación y pruebas varias veces, madurando el sistema en cada iteración. Las etapas que se muestran en los siguientes puntos, son las versiones finales de las mismas.

4.1 Análisis

La estrategia a seguir se basa en centrarse al máximo en cloud, coste cero y apoyarse al máximo en herramientas estables ya existentes, generando una base lo más sólida posible. Las ventajas de esta estrategia son todas las intrínsecas del cloud y del uso de las herramientas las herramientas mencionadas en el estado del arte. Los inconvenientes son las heredadas igualmente del cloud y el uso de herramientas de terceros más algunas limitaciones por el uso de versiones gratuitas y el coste de búsqueda y obtención de los conocimientos necesarios para llevar la estrategia a cabo.

4.1.1 Herramientas

En primer lugar y después de tener el conocimiento del estado del arte, se deciden qué herramientas son las idóneas para resolver los objetivos principales del proyecto así como ver que opciones serían viables para desarrollar.

En primer lugar, se escoge Python como lenguaje de programación, dada su versatilidad y cantidad de herramientas para el procesado de datos tales como la librería pandas [10]. Para una mayor claridad y portabilidad del código, se utiliza Jupyter Notebook [11] desde la plataforma de ciencia de datos Anaconda [12]. El código generado puede correr tanto *on-premise*, en una instalación de Anaconda habitual, como en cloud, en una plataforma como Databricks Community [13].

El almacenamiento de los datos procesados se produce en *Google Drive* [14]. *Google Drive* ofrece 15 GB de almacenamiento gratuito y multitud de *API*s y herramientas para el acceso y modificación de los datos desde prácticamente cualquier sitio. De este modo desde el código

de Python se puede enlazar con *Google Drive* para publicar los datos en formato excel y ofrecerlos mediante la *API* de *Google Sheets* (Al almacenarse los ficheros como excel en *Google Drive*, también son accesibles desde la *API* de *Google Sheets*, lo que nos evita las limitaciones de las propias hojas de cálculo de Google y nos sigue permitiendo las ventajas de la *API* de *Google Sheets*) hacia la *Tableau Public* , la herramienta de visualización.

Como herramienta de visualización se utiliza *Tableau Public*. Este permite la conexión con *Google Sheets* y la sincronización de datos con el mismo. Las visualizaciones se pueden configurar de forma *responsive*, lo que permiten un formato de visualización diferente según el tipo de dispositivo desde donde se muestran. Estas visualizaciones son públicas, y se pueden incrustar mediante código HTML en cualquier página web.

Como presentación de la información al usuario se realiza una web, también *responsive*, donde se estructura la información y se incrustan las visualizaciones de *Tableau Public*.

4.1.2 Ciclo de vida de los datos

El ciclo de vida de los datos de este proyecto sigue varias etapas. Comienza en la descarga de datos desde la *clinicaltrials.gov* y acaba por percibirlos un usuario final en una página web (ambientada como un Dashboard). La primera idea que surge es la frecuencia de refresco y la metodología de actualización. En cuanto a la frecuencia de actualización una actualización diaria es razonable. Actualizar los datos puede ser un proceso muy complejo si se intenta mantener un versionado de los mismos, de modo que se prevé descargar de nuevo, procesar y publicar todos los datos en cada actualización de los mismos. De este modo, las etapas resultantes son:

1. *ETL*. Se extraen, se transforman a un formato adecuado y cargan los datos en el sistema.
    a. Transformación de datos. Transformaciones y procesado de los datos según los objetivos de visualización. Esta etapa se ejecuta incrustada en la *ETL.*
    b. Se incluyen procesos de limpieza de datos.
2. Publicación de los datos. Los datos resultantes se publican de forma que la capa de visualización disponga de acceso a ellos.
3. Visualizaciones. Los datos son utilizados para realizar visualizaciones que contesten las preguntas que se han planteado. Estas visualizaciones se diseñan en local pero una vez publicadas se gestionan por completo en la nube. Además el sistema de visualización actúa como caché independiente de los datos, mostrando los últimos datos cacheados. Posteriormente estas visualizaciones son incrustadas en una página web.
4. Los datos persisten hasta que se refresca todo el proceso en cada actualización. Todos los datos anteriores son reemplazados por los nuevos. No se almacena control de

versiones en los datos. (La actualización pretende ser diaria, y por tanto los datos más recientes visualizables serán los datos hasta el dia anterior).

4.2 Diseño

Fuente de datos

Clinicaltrials.gov es la fuente de datos seleccionada. Los datos son descargados de forma particionada por el proceso ETL.

ETL

Extraer, transformar y cargar datos en el almacenamiento de datos.
Este proceso puede ejecutarse utilizando un *notebook* de Python tanto en la nube como en local. La Databricks Community proporciona *notebooks de* Python en la nube gratuitos y decentes.

Almacenamiento de datos

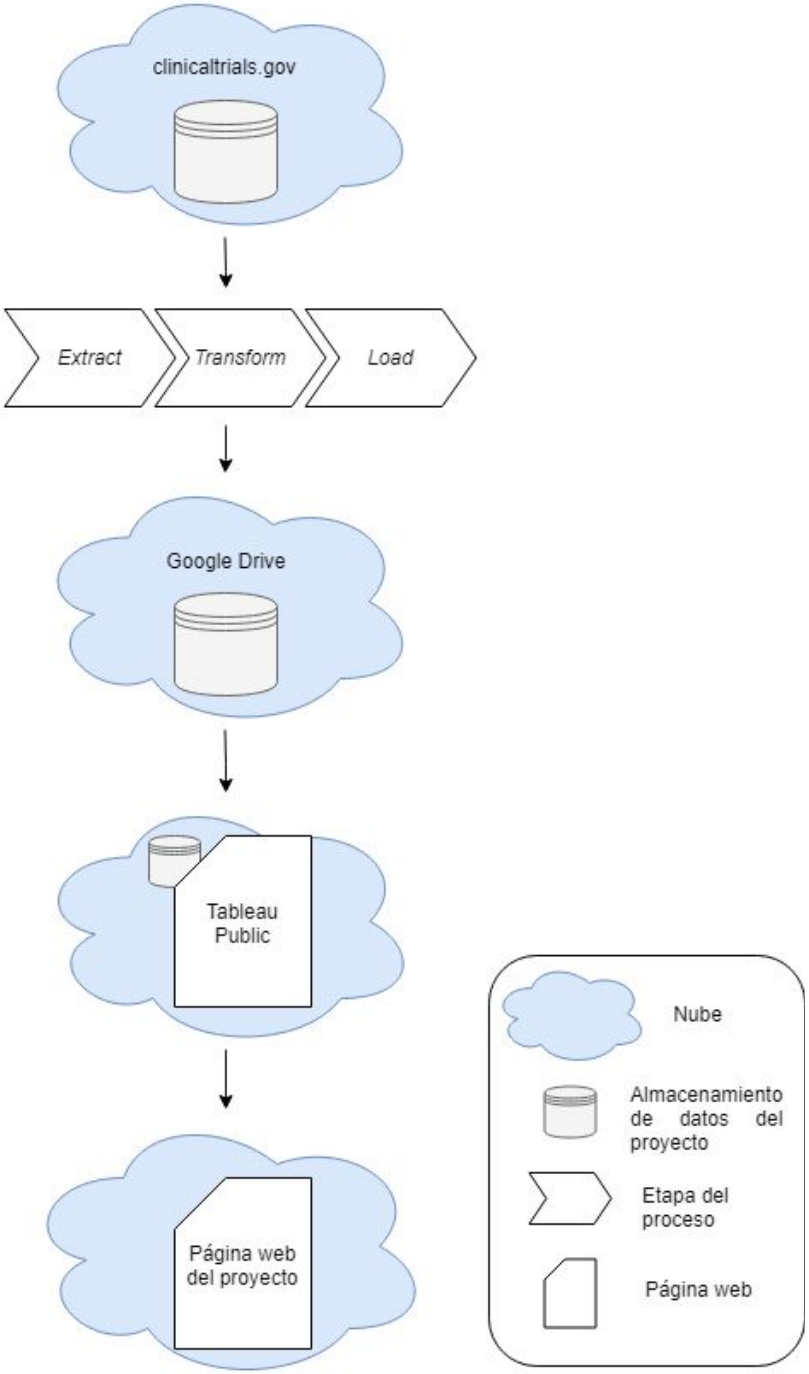
Google Drive tiene 15 GB libres de almacenamiento de datos y muchas características como la aplicación de operaciones CRUD desde cualquier lugar, incluidos los *notebooks* de Python.

Sistema de visualización

Tableau Public es una herramienta de visualización que proporciona 10 GB de almacenamiento en la nube gratuito y muchas conexiones de datos, como la conexión sincronizada de Google Sheets.

Presentación

Los datos se presentan a los usuarios en una página web del proyecto. Hay hospedajes gratuitos como infinityfree.net que proporciona suficientes funciones para el proyecto de costo 0.



Arquitectura del sistema. Herramientas y procesos

4.3 Programación

Tal y como se ha mencionado anteriormente, se utiliza Python como lenguaje de programación. Siguiendo la metodología de prototipage se itera sobre el código sobre un *Notebook.* Este ha pasado por diversas versiones generales y subversiones (mejoras y correcciones menores) antes de alcanzar la definitiva. El objetivo final es; la generación de un proceso de extracción de los datos de registros clínicos de *clinicaltrials.gov*, la aplicación de diferentes procesos de transformación de datos para generar ficheros de datos focalizados un objetivo de visualización, y la carga de estos datos al repositorio de datos.
En la primera versión del código, se descargaba un único fichero con todos los registros clínicos y se intentaba procesar de una sola vez lo cual produce una saturación de capacidad de la memoria RAM.
En siguientes versiones se alterna la descarga por bloques contra la descarga unitaria, el procesado por bloques contra el unitario, y el procesado multi núcleo contra el procesado en un único núcleo.
Tras las convenientes pruebas se encuentra que el uso de múltiples núcleos para el procesado no es viable, pero si la descarga de ficheros por bloques a la vez que un procesamiento por bloques, corran o no los procesos en el mismo núcleo, dado que principalmente consumen recursos diferentes.
Finalmente se encuentra una forma de compatibilizar el procesado con bloques con la generación de un único fichero por objetivo de visualización que contenga el conjunto de datos total generado a partir de los bloques.
En la versión definitiva el tiempo de ejecución de la *ETL* conseguido sobre un ordenador corriente (3,6 Ghz) es de alrededor de 3h (dato orientativo), con un consumo de RAM de alrededor de 1GB (normalmente inferior), lo cual se encuentra muy por debajo de los 6GB establecidos como referencia por *Databricks Community*. El tamaño de los ficheros excel resultantes no supera un Gigabyte. No obstante, el espacio en disco requerido para la ejecución del proceso es de varios Gigabytes (recomendado disponer de 16Gb, como dato orientativo). El tiempo de ejecución es el mejor conseguido durante las pruebas, siendo viable la actualización de hasta una frecuencia diaria de los datos mediante la ejecución del mismo. Adicionalmente, el sistema es capaz de ir ampliando los datos, generando nuevas hojas en el fichero excel resultante cuando se exceda el límite de filas por hoja, mientras que el motor de visualización es capaz de interpretar las nuevas hojas como más datos de la misma tabla a mostrar, todo esto sin intervención humana.

La parte de desarrollo de las visualizaciones de Tableau se ha realizado con una licencia de estudiante de *Tableau Desktop* [15] y publicado las visualizaciones (sincronizadas con Google Sheets) en *Tableau Public*. En la *ETL* los datos han sido preparados adecuadamente para su visualización.

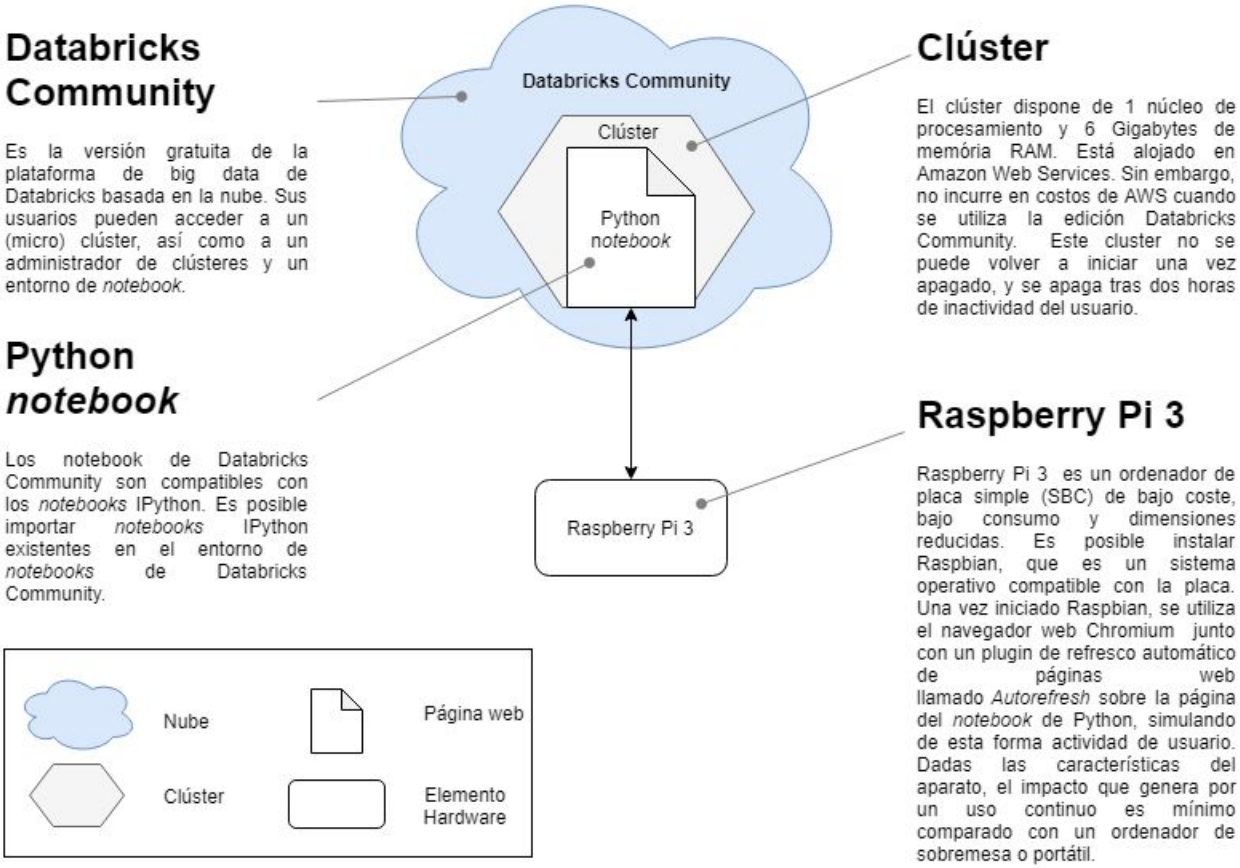
La parte de desarrollo web *responsive* se ha realizado mediante el uso de herramientas WYSIWYG [16], después de una investigación para la elección de las tecnologías. Además se

utilizan diferentes sistemas de optimización, como la caché. No obstante, las limitaciones del hosting gratuito se hacen visibles, sobretodo, en la primera carga de la página.

4.3.1 Problemáticas encontradas

Databricks

Para seguir con la línea de utilizar todo en cloud, se comenzó a desarrollar el sistema en Databricks Community (después de una fase de búsqueda de plataforma cloud), que ofrece un cluster de 1 core y 6GB de RAM. La primera problemática es que el cluster se apaga tras una hora de inactividad, entendiendo como inactividad la no interacción del usuario con el clúster. No importa si hay algún proceso corriendo, el clúster se apaga, no se puede volver a arrancar y se pierden todos los datos almacenados y la configuración del mismo teniendo que crear uno nuevo. Lo único que se conserva es el código en el *Notebook*. No obstante, es posible solucionar este problema con un plugin de refresco automático del navegador web *Google Chrome* llamado *Auto Refresh*. Para mayor autonomía se puede instalar el plugin en el *Chromium* que viene por defecto en el sistema operativo *Raspbian* de una *Raspberry Pi*, y de esta forma dar a entender al cluster que lo estás utilizando. El concepto se ilustra en el siguiente diagrama:



Uso de Raspberry Pi para evitar el apagado por inactividad del clúster

El segundo problema, es la inestabilidad del clúster. Tras algún exceso de la capacidad de memoria RAM o tras alguna excepción de código, el clúster puede dejar de ser operativo teniendo que crear otro nuevo. Este problema descarta el uso del *Databricks Community* como plataforma de desarrollo para este proyecto, ya que se trata con ficheros de texto que en disco superan los 6 GB (~6.5GB). No obstante seguiría siendo posible el despliegue de un código estable de bajo consumo de memória RAM.

Tiempo de investigación, tiempos de espera y gestión de memoria

Al tratar con una cantidad tan grande de datos, surgen multitud de problemáticas asociadas al big data y deja de ser posible, al menos en algunos casos, la idea de seleccionar una muestra de datos y probar el código con ella, para luego hacerlo funcionar con el conjunto completo de datos. Entran en juego multitud de factores como la memoria disponible, la capacidad de las librerías utilizadas, los tiempos de espera para pruebas de datos masivos, etc.
Encontrar la forma de segmentar por bloques la entrada de datos, el procesado, y el grabado de datos es una tarea requerida y una de las más costosas. Por ejemplo, las librerías de escritura de excel incorporadas en *Pandas*, simplemente no son capaces de ir añadiendo datos a un fichero excel ya existente sin cargarlo por completo en memoria. Una posible solución para el caso de la librería *xlsxwriter* es generar manualmente un fichero solo escritura en un modo específico de uso de memoria constante indicado con el parámetro *constant_memory*, donde la librería irá generando archivos temporales que ocupan alrededor de 5 veces más tamaño que en memoria (dato orientativo), para finalmente unirlos y generar el fichero final tras cerrar la escritura de datos (siempre que no se sobrepase el límite de filas por hoja). Todo esto requiere de decenas de pruebas e investigación.
A continuación se destacan las conclusiones de algunas de las pruebas o estudios realizados:

- Uso de *Google Sheets* como formato almacenamiento de datos. Utilizar las hojas de cálculo de Google en principio simplifica la transferencia de datos entre el *notebook* de Python y el almacenamiento de datos en *Google Drive*, con la posibilidad de enviar los datos a la hoja de cálculo en la nube a medida que se van generando. Existen librerías de Python como *gspread* que operan directamente contra la *API* de *Google Sheets*. Se encontraron limitaciones en el número de filas y columnas por hoja y *Google Sheets* dejó de cumplir con las expectativas. No obstante, en las pruebas realizadas sólo se consideraba el uso de una sóla hoja, cuándo más adelante se observó que el uso de múltiples hojas en el mismo libro también es totalmente compatible con el motor de visualización. A pesar de esto, quedan pendientes de comprobar otras posibles limitaciones o inconvenientes para que realmente esta sea la mejor opción, como el tamaño total por documento, la compatibilidad para exportación de los datos a un fichero local, el mantenimiento de posibles cambios en la *API*, la necesidad de conexión ininterrumpida a internet durante el proceso de datos, la ausencia de datos en local para posibles análisis a posteriori, etc. Por lo tanto, aunque finalmente no se utilice, sigue siendo otra opción que podría ser viable.

- El uso de CSV como formato de almacenamiento de datos. Mientras que otros formatos como XLSX produce archivos binarios y su manipulación es más compleja, un archivo en formato CSV contiene texto plano con valores separados por comas, por lo que se puede manipular de la misma forma que cualquier otro archivo de texto plano, con un coste de memoria constante y sin requisitos especiales como la carga por completo del mismo en memoria para realizar una escritura de datos, tal y como pasa en los archivos en formato XLSX. Por otro lado, cualquier fallo en la estructura de caracteres delimitadores provoca que las columnas de la fila afectada se desplacen, afectando a la integridad de los datos. A esto se suma que CSV no es un formato soportado por la *API* de *Google Sheets*, por lo que la sincronización de datos con el motor de visualización no es posible.
- Uso de base de datos relacional. El uso de una base de datos aporta ventajas y desventajas similares a utilizar *Google Sheets* en cuanto a sencillez, consumo de memória, integridad de los datos y la compatibilidad con el motor de visualización, además de no sufrir las limitaciones de número de filas o columnas. Por otro lado sigue dependiendo de una conexión estable durante todo el proceso *ETL*, e implica otras problemáticas relacionadas con la eficiencia y actualización de datos. Existen bases de datos relacionales dedicadas en la nube, pero habitualmente se encuentran en las capas de prueba gratis por periodo de tiempo limitado [17]. Otra posible opción es utilizar la base de datos de algún hosting web gratuito, pero estas sufren de limitaciones de espacio, uso o rendimiento que eventualmente impedirían la actualización de los datos. Por lo tanto no se han llegado a realizar pruebas de uso.

Multicore

El uso de múltiples núcleos del procesador en Jupyter, al menos por ahora [18], no ha sido una opción viable. Incluso utilizando la librería multiprocessing sólo se utiliza un núcleo, siguiendo con la limitación de GIL[18], el cual es un tipo de bloqueo de proceso que Python utiliza cuando trata con procesos. En general, Python solo usa un hilo para ejecutar el conjunto de declaraciones escritas. Esto significa que en python solo se ejecutará un hilo a la vez. El rendimiento de un proceso de un único hilo y un proceso con múltiples hilos será el mismo en python y esto se debe a GIL. No podemos lograr multihilo en python porque tenemos un bloqueo de intérprete global que restringe los hilos y funciona como un solo hilo.

Errores y límites en las librerías y formatos

La librería *pandas* proporciona los objetos *Dataframe*, los cuales son tablas de datos con gran funcionalidad. Estos *Dataframe* tienen diferentes funciones. Una de ellas es la opción de iterar filas, que devuelve el índice de la fila y la fila en cuestión. Al llenar la tabla con algunos datos que se deben de salir de las expectativas de la tabla, la función devuelve 0 como índice. Algo similar o quizás heredado del mal funcionamiento del *Dataframe* pasa al guardar la tabla en excel.

Otro caso que ha sucedido es que diferentes funciones o formas de obtener un valor, pueden dar resultados diferentes, como sucede con el número de columnas. Contar el número de

columnas de la lista de columnas a veces tiene un resultado diferente que obtener el número de columnas del molde de la tabla.
Algunas de estas cuestiones no se llegan a profundizar y simplemente se busca una solución que funcione.
Por lo general, utilizar cada función con una única responsabilidad, y ser consistente con el uso de las variables y de dónde se extraen, suele solucionar los problemas de inconsistencia que a priori no se deberían dar.
Por otro lado *Microsoft Excel* tiene un límite de 1.048.576 (filas) x 16.384 (columnas) o 33.554.432 celdas por hoja. Los programas y librerías que utilizan el formato xlsx de *Microsoft Excel* implementan estos límites (en *xlsxwriter* no se permiten intercambiar filas por columnas) para asegurarse de que los ficheros podrán ser leídos por el software de Microsoft Excel. Al expandir algunos datos del conjunto de datos del proyecto se supera la limitación de filas, lo que fuerza utilizar múltiples hojas o cambiar de formato. Otros posibles formatos como CSV no son aceptados por la *API* de *Google Sheets*, la cual es esencial para la actualización automática. Por tanto, se han de gestionar los datos en múltiples hojas en formato Excel.

4.3.2 Pruebas y sobrecarga.

Siguiendo con el modelo de prototipage, estas dos etapas han sido incluidas en la etapa de Programación, utilizando todo el tiempo de sobrecarga en desarrollo. Las pruebas han sido realizadas tanto en la ejecución del código original
(pruebas de integración), como en pruebas unitarias de las funciones individuales en otros libros de trabajo temporales. Estas pruebas han sido realizadas a modo *ad hoc* según han sido requeridas para comprobar las funciones que se han realizado, sin seguir una metodología TDD[19].

4.4 Elementos de la aplicación.

4.4.1 ETL

El código del proceso *ETL* escrito en Python se entrega en formato .ipynb y .html en la carpeta "ETL", y se encuentra disponible en el Anexo 1 de este documento.
Las decenas de pruebas realizadas en otros notebooks se omiten ya que el fichero entregado es el resultado funcional, mientras que las pruebas son complementarias y no se encuentran preparadas para ser presentadas.

4.4.2 Visualizaciones

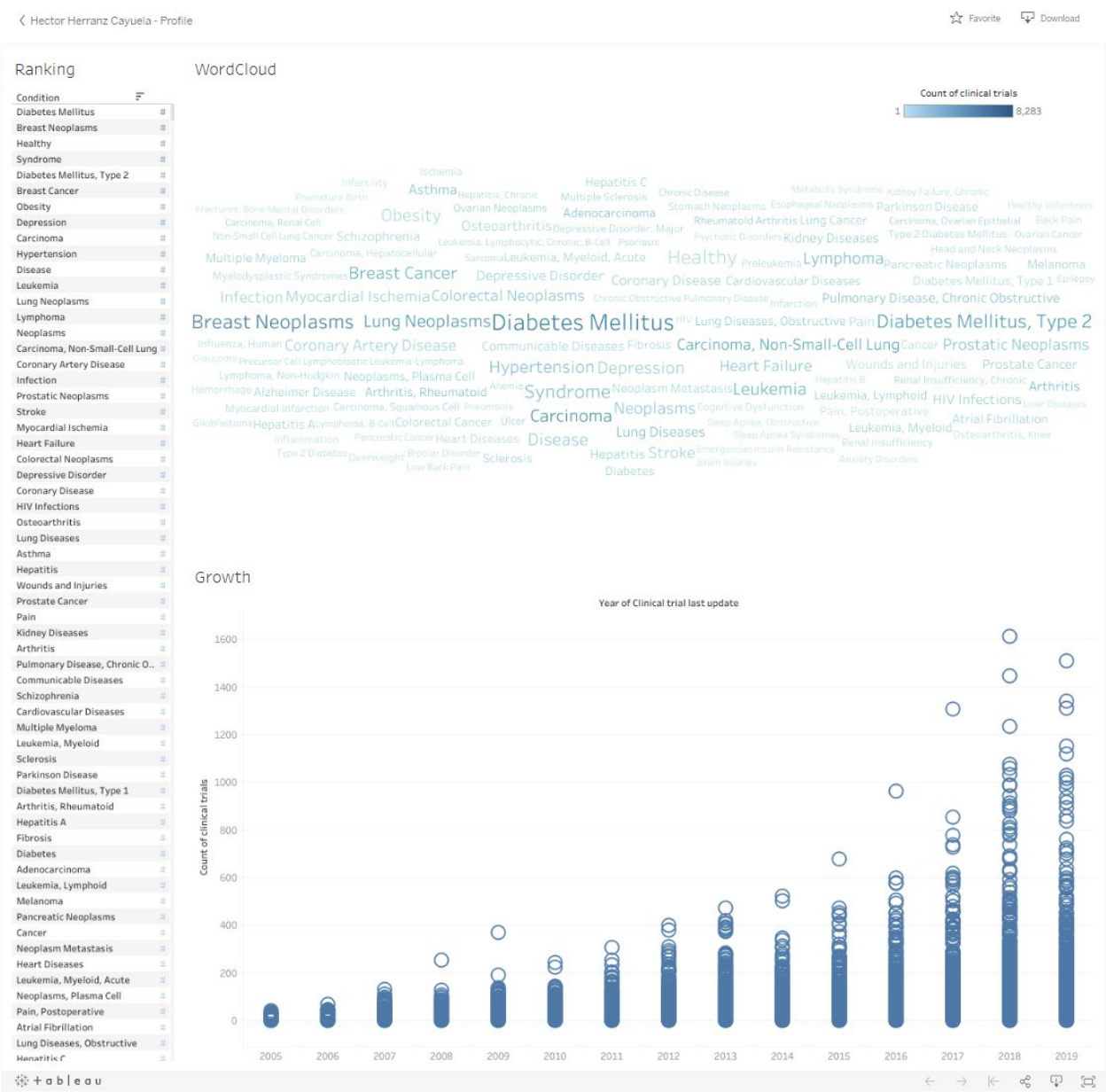
Las visualizaciones se encuentran disponibles en el siguiente enlace:
https://public.tableau.com/profile/hector.herranz.cayuela#!/

Las *Vizzes* que corresponden al proyecto son las que empiezan con el código de nombre QX_ ó qX_ donde X es un número.
Adicionalmente se adjuntan bajo la carpeta "Visualizaciones".
En el momento de la entrega utilizan datos parciales en el fichero de visualización. No se adjuntan los extractos de datos, ya que la conexión se establece con una cuenta de Google utilizada para este proyecto. Los datos de las visualizaciones del enlace proporcionado (Tableau Public) pueden sufrir actualizaciones dada la estrategia de actualización definida.

Al incrustar las visualizaciones en la página web del proyecto, estas se ajustan al tamaño de la pantalla y cambian la distribución de elementos según el dispositivo desde donde se visualizen. En cambio, al visualizarse directamente desde Tableau Public se quedan en un tamaño fijo con orientación vertical, que no es para lo que han estado diseñadas. No obstante, a continuación se muestran las visualizaciones en *Tableau Public*.

q1_studiedDiseasesNow

Enlace:
https://public.tableau.com/profile/hector.herranz.cayuela#!/vizhome/q1_studiedDiseasesNow/Dashboard1
Pregunta que resuelve:
¿Qué enfermedades son las más estudiadas hoy en día?
Orientación:
Conocer la tendencia de estudio de las enfermedades.



q1_studiedDiseasesNow

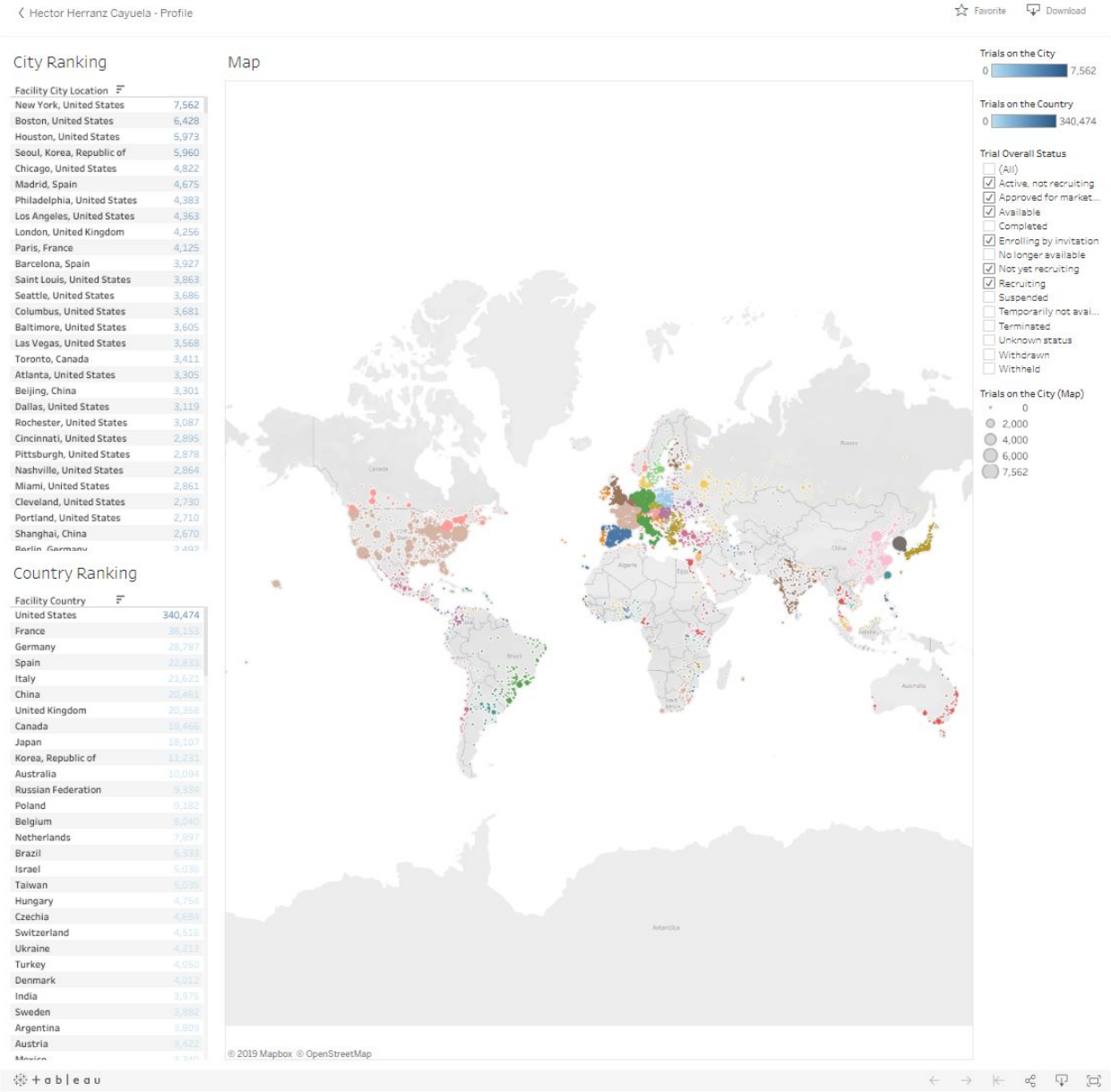
En esta visualización se muestra un ranking, una nube de palabras y un gráfico de burbujas. El ranking conceptualmente muestra de forma descendente las enfermedades (bajo el nombre *condition*) más estudiadas.
La nube de palabras es una representación visual de las enfermedades que superan un cierto umbral de número de estudios. Finalmente el gráfico de burbujas da una idea del crecimiento y la posición actual de las enfermedades.
Es una visualización interactiva y seleccionado una enfermedad en cualquier gráfico, se resaltará su posición en los demás.

Q2_countriesStudiesInProgress

Enlace:
https://public.tableau.com/profile/hector.herranz.cayuela#!/vizhome/Q2_countriesStudiesInProgress/Dashboard1

Pregunta que resuelve:
¿Qué países tienen más estudios en curso hoy en día?

Orientación:
Conocer el interés de los países sobre los estudios clínicos.



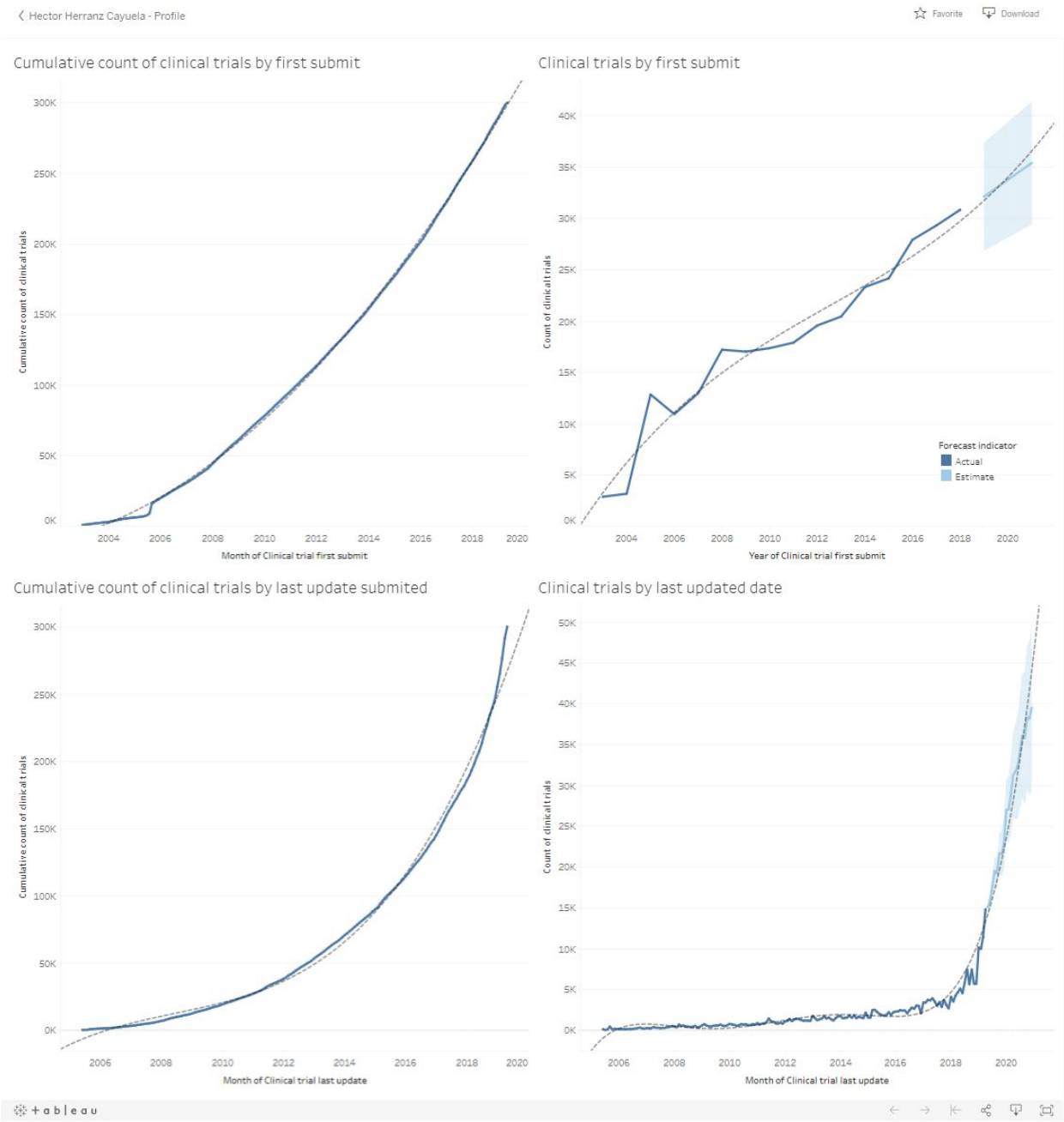
Q2_countriesStudiesInProgress

En esta visualización se muestra un ranking por ciudad, un ranking por país y un mapa de burbujas por ciudad.

El ranking por ciudad muestra de forma descendente las ciudades con más estudios clínicos en curso.
El ranking por país muestra de forma descendente los países con más estudios clínicos en curso.
El mapa muestra burbujas el tamaño de las cuales va en relación con el número de estudios clínicos. El color utilizado diferencia las burbujas de diferentes países.
En el lateral derecho se muestra una lista de casillas de verificación que dan la opción de filtrar por otros criterios de estado de los estudios.
Es una visualización interactiva y seleccionado una elemento en cualquier gráfico, se resaltará su posición en los demás.

Q3_evolutionNumberTrials

Enlace:
https://public.tableau.com/profile/hector.herranz.cayuela#!/vizhome/Q3_evolutionNumberTrials/Dashboard1
Pregunta que resuelve:
¿Cómo ha evolucionado el número de estudios clínicos a lo largo del tiempo?
Orientación:
Conocer la evolución del número de estudios clínicos.



Q3_evolutionNumberTrials

En esta visualización se muestran cuatro gráficos lineales; un gráfico lineal acomulativo y uno con predicción de valores, ambos contando los estudios clínicos por fecha de inicio, y otra pareja de gráficas iguales pero esta vez contando estudios clínicos por fecha de última actualización de los mismos.
En todas las gráficas se muestra una línea de tendencia en color gris.

Q4_conditionEvo

Enlace:

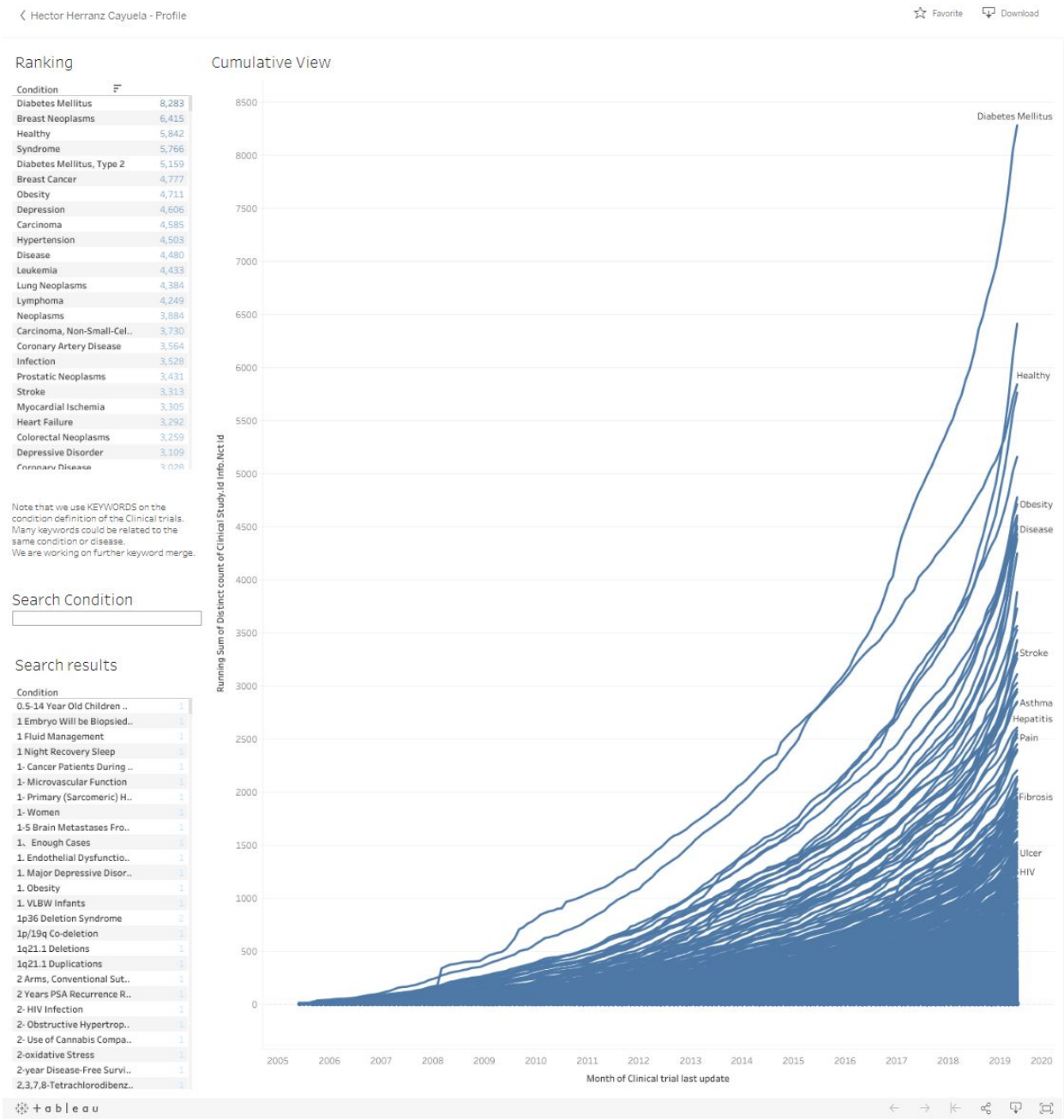
https://public.tableau.com/profile/hector.herranz.cayuela#!/vizhome/Q4_conditionEvo/Dashboard1

Pregunta que resuelve:

¿Cómo ha evolucionado el número de estudios clínicos de una enfermedad?

Orientación:

Conocer la tendencia de estudio de cada enfermedad.



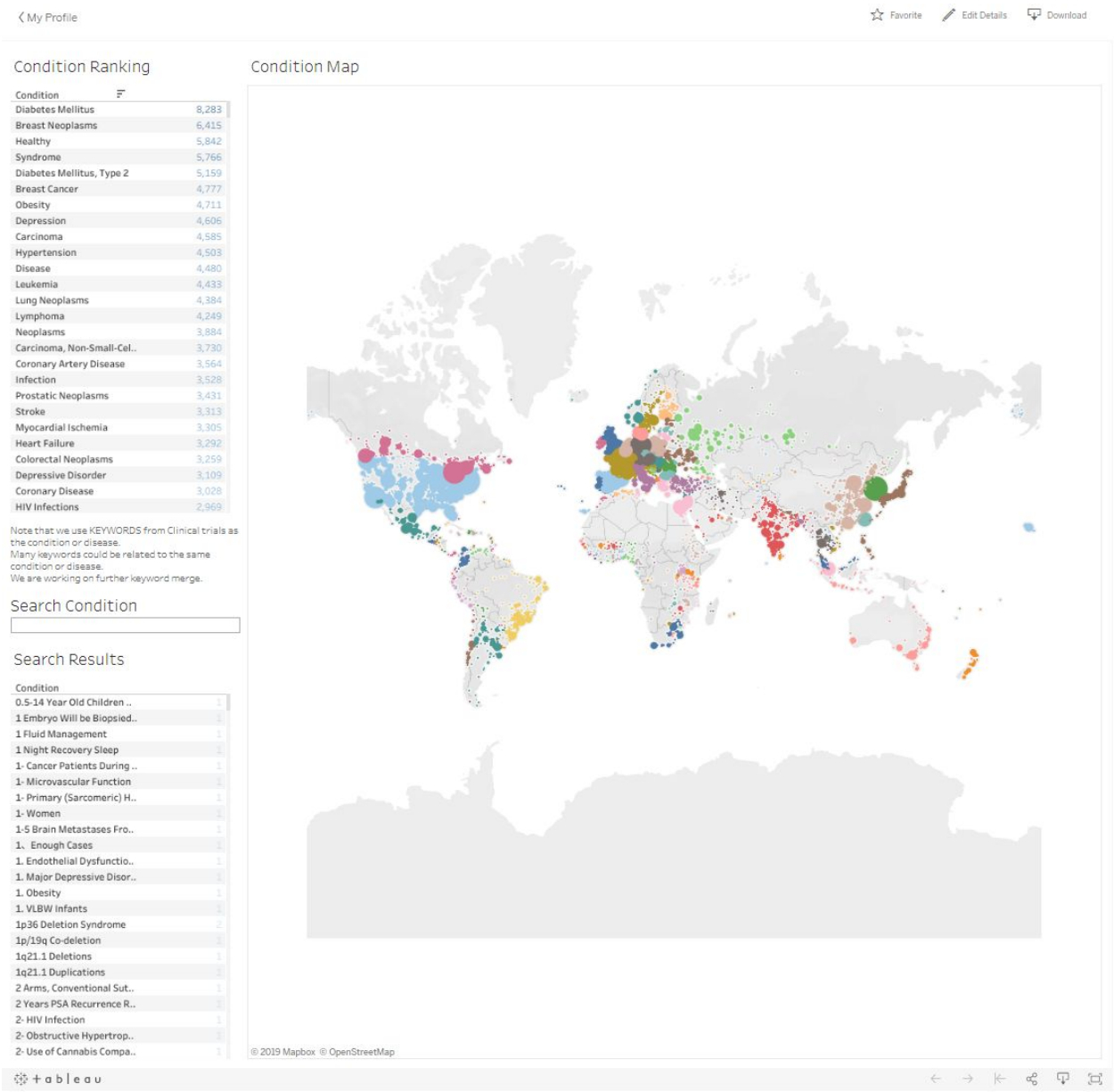
Q4_conditionEvo

En esta visualización se se muestra un ranking, una caja para buscar una enfermedad, una lista de resultados de búsqueda y una grafica lineal acomulativa.

El ranking muestra de forma descendente las enfermedades que cuentan con más estudios clínicos.
En la caja de búsqueda se permite buscar por enfermedades, las cuales se mostrarán en la lista de resultados.
En la grafica lineal acomulativa se muestra una línea por cada enfermedad.
Es una visualización interactiva y seleccionado una elemento en cualquier gráfico, se resaltará su posición en los demás.

Q5_conditionWhere

Enlace:
https://public.tableau.com/profile/hector.herranz.cayuela#!/vizhome/Q5_conditionWhere/Dashboard1
Pregunta que resuelve:
¿Dónde se realizan los estudios clínicos sobre una enfermedad?
Orientación:
Conocer el interés de las ciudades por una enfermedad.



Q5_conditionWhere

En esta visualización se muestra un ranking por enfermedades, una caja de búsqueda de enfermedades, una lista de resultados de búsqueda y un mapa de burbujas por ciudad.
En el ranking se muestra de forma descendente las enfermedades con más estudios clínicos.
En la caja de búsqueda se permite buscar por enfermedades, las cuales se mostrarán en la lista de resultados.
El mapa muestra burbujas el tamaño de las cuales va en relación con el número de estudios clínicos. El color utilizado diferencia las burbujas de diferentes países.
Es una visualización interactiva y seleccionado una elemento en cualquier gráfico, se resaltará su posición en los demás.

Q6_conditionInvestigator

Enlace:

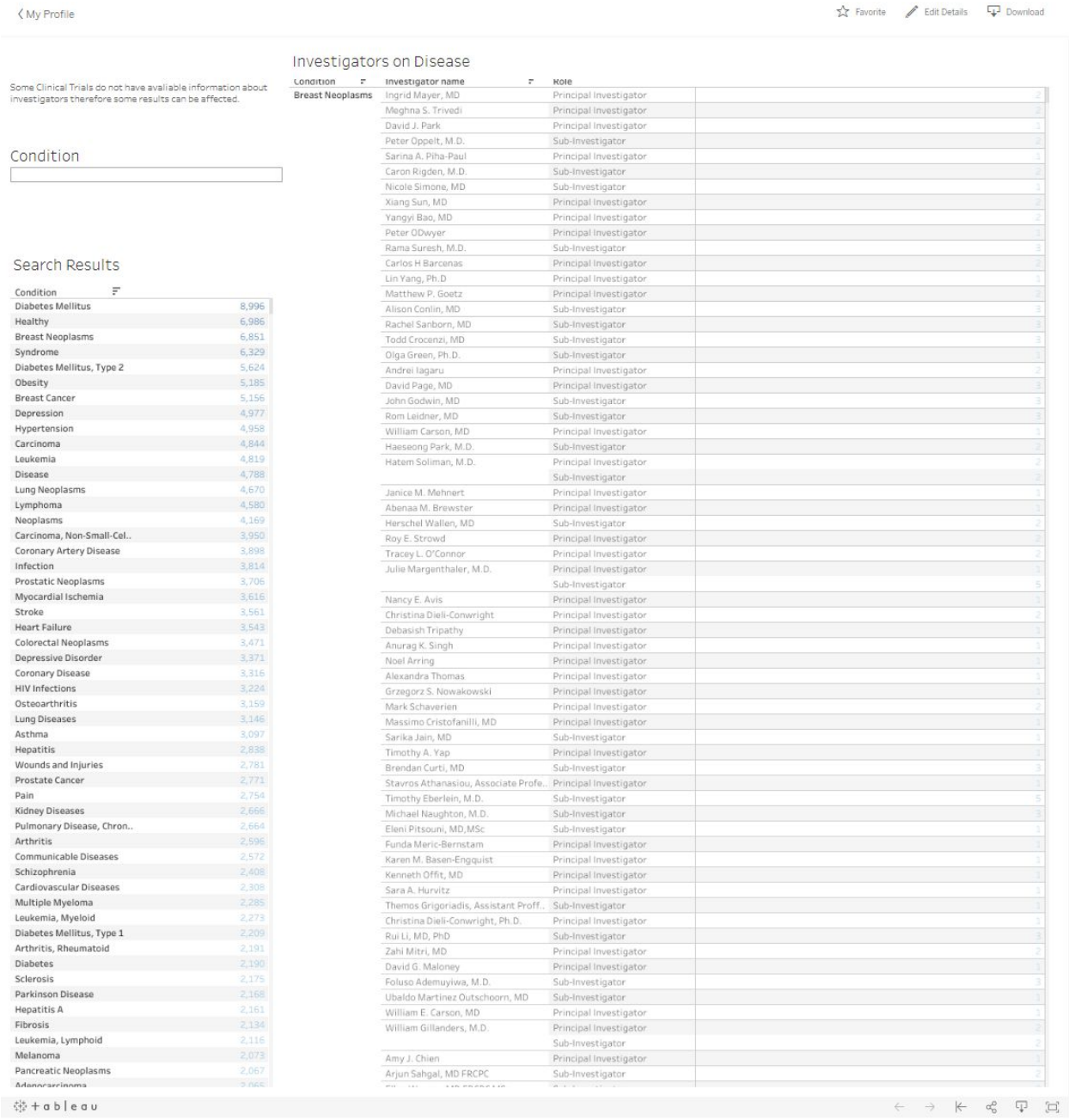
https://public.tableau.com/profile/hector.herranz.cayuela#!/vizhome/Q6_conditionInvestigator/Dashboard1

Pregunta que resuelve:

¿Cuántos estudios clínicos ha realizado cada investigador sobre una enfermedad?

Orientación:

Conocer los investigadores de una enfermedad.

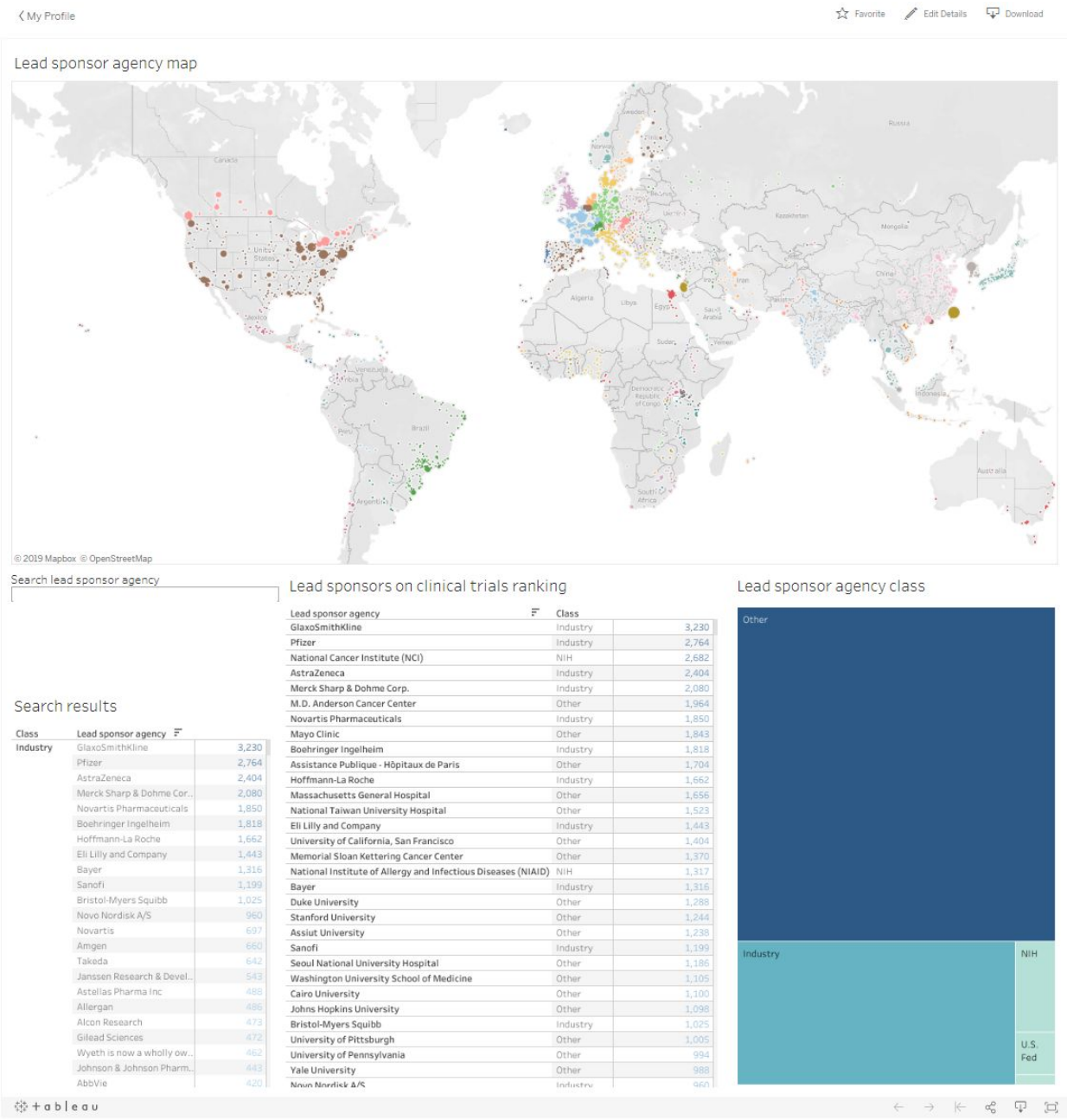


Q6_conditionInvestigator

En esta visualización se muestra una caja de búsqueda por enfermedad, una lista de resultados de búsqueda y una tabla de datos de investigadores. Es una visualización interactiva y seleccionado una elemento en cualquier gráfico, se resaltará su posición en los demás.

Q7_leadSponsorAgencies

Enlace:
https://public.tableau.com/profile/hector.herranz.cayuela#!/vizhome/Q7_leadSponsorAgencies/Dashboard1
Pregunta que resuelve:
¿Qué entidades están patrocinando más estudios clínicos?
Orientación:
Conocer los principales patrocinadores.



Q7_leadSponsorAgencies

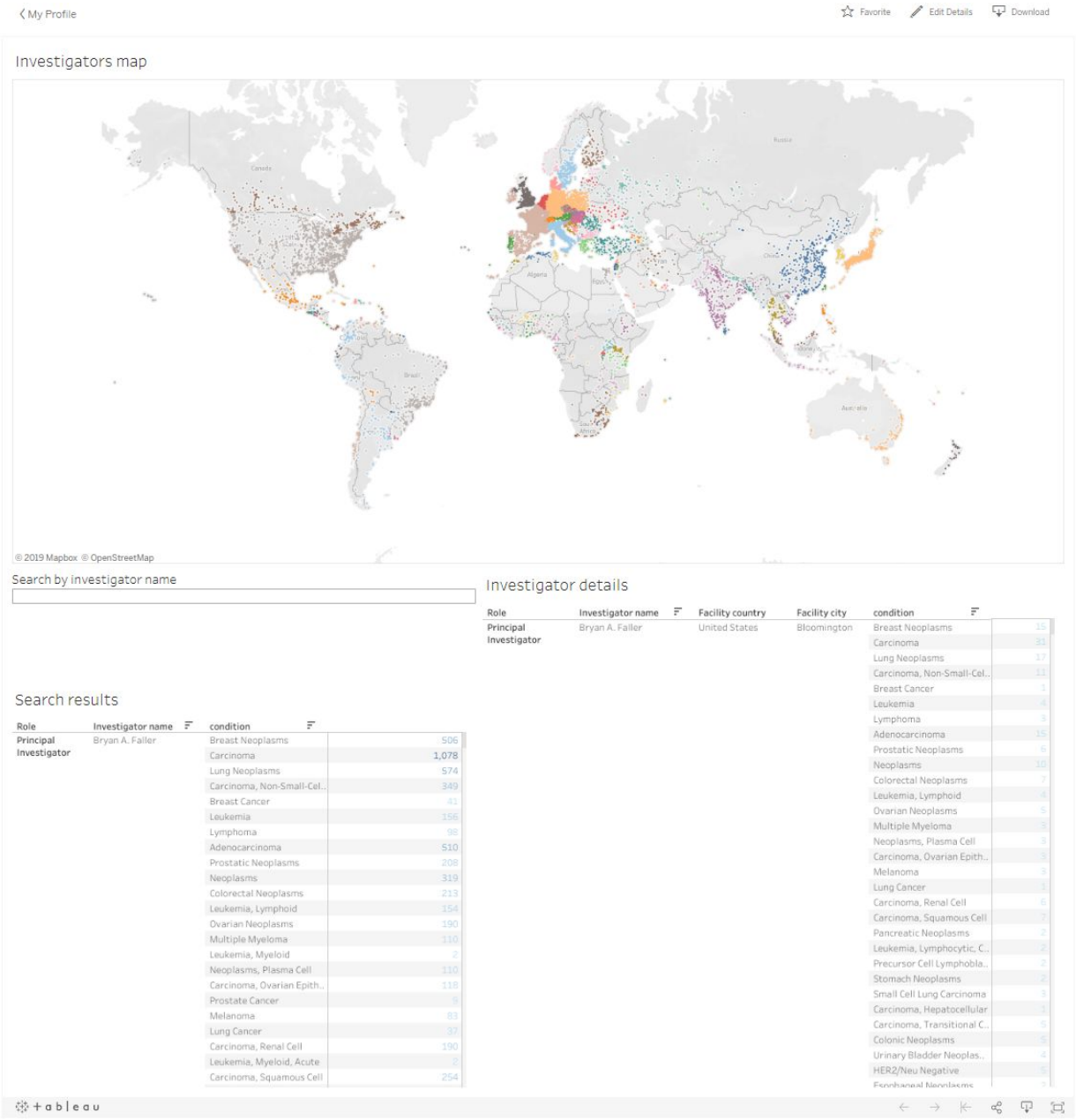
En esta visualización se muestra un mapa de burbujas por patrocinador dónde el tamaño de la burbuja crece según el número de estudios que ha patrocinado, una caja de búsqueda por patrocinador, una lista de resultados de búsqueda, una tabla de información de sponsors y un *treemap* de los tipos de patrocinadores.
Es una visualización interactiva y seleccionado una elemento en cualquier gráfico, se resaltará su posición en los demás.

Q9_investigatorsWork

Enlace:
https://public.tableau.com/profile/hector.herranz.cayuela#!/vizhome/Q9_investigatorsWork/Dashboard1

Pregunta que resuelve:
¿En qué otros temas trabaja un investigador?

Orientación:
Conocer investigadores



Q9_investigatorsWork

En esta visualización se muestra un mapa de burbujas por investigador donde el tamao de la burbuja crece según el número de estudios clínicos que ha realizado un investigador, una caja de búsqueda de investigadores, una lista de resultados de búsqueda y una tabla de información de investigadores.
Es una visualización interactiva y seleccionado una elemento en cualquier gráfico, se resaltará su posición en los demás.

4.4.2 Web

La web se ha realizado en inglés y en el momento de la entrega se encuentra visible en el siguiente enlace:
http://www.clinicaltrialsintelligence.epizy.com/
No obstante, se prevé que el proyecto siga adelante y por tanto es posible que cambie de dominio y/o hosting.
Adicionalmente se adjunta la última copia de seguridad disponible de la web bajo la carpeta "Web".
La web puede tardar en cargar en la primera apertura a pesar de las optimizaciones realizadas, dadas las limitaciones del hosting gratuito.
Se adjuntan capturas de pantalla de la web del proyecto en el Apéndice 2, mostrando las diferentes visualizaciones y un ejemplo de cómo se adapta según el dispositivo desde donde se visualice.

5. Resultados y conclusiones.

Se ha conseguido encontrar una fuente de datos confiable y alcance mundial que proporciona datos abiertos sobre estudios clínicos en diferentes formatos y que habilita diferentes métodos de obtención de los mismos, cumpliendo con los requisitos de este proyecto.
A su vez, se ha conseguido desarrollar con èxito un proceso *ETL* que puede ser ejecutado tanto en local como en la nube y que es capaz de; actualizar los datos hasta en una frecuencia diaria, soportar el futuro crecimiento de la cantidad de datos disponibles desde la fuente de datos y la carga de los mismos en un repositorio de datos en la nube.
Se ha conseguido hallar e incorporar con éxito un repositorio de datos de acceso libre en la nube es capaz de recibir los datos del proceso *ETL*, almacenarlos y servirlos al motor de visualización en la nube de forma compatible, síncrona y eficiente desde el punto de vista de este proyecto.
Se ha conseguido hallar e incorporar con éxito un motor de visualización de acceso libre en la nube es capaz de sincronizarse de forma periódica con el repositorio de datos en la nube, cargar los datos en caché, y mostrarlos a través de visualizaciones interactivas, incrustables en formato web y adaptables al dispositivo desde donde se visualizen.
Adicionalmente se a generado mediante servicios gratuitos una web dedicada al proyecto que presenta de forma amigable al usuario cada pregunta que plantea sobre los estudios clínicos y su respuesta en forma de visualización de datos.

El proyecto se ha conseguido realizar utilizando las versiones gratuitas de todas las herramientas y servicios utilizados. No obstante, hay algunos componentes del proyecto que, desde un punto de vista de usuario final, degradan notablemente la calidad del mismo. Estos componentes son la versión gratuita del motor de visualización y del servicio de hospedaje web, que disponen de bajos recursos y por lo tanto los tiempos de carga son inusualmente elevados. Este problema puede solucionarse obteniendo versiones de pago de los mismos las cuales permitan aumentar los recursos disponibles y agilizar los tiempos de carga tanto de las visualizaciones de datos como los de la web, lo que incrementa considerablemente la experiencia de usuario.

No se han encontrado otros puntos críticos que resaltar. No obstante quedan algunas cuestiones técnicas de mejora a evaluar como la utilización de *Google Sheets* en lugar de ficheros xlsx, tal y como se explica en la sección 4.3.1.
Tampoco existen a priori limitaciones legales directas para la posible comercialización del proyecto, una vez utilizando las versiones de pago de las herramientas y servicios.

Considerando el proyecto como una prueba de concepto, se han obtenido los resultados esperados, cumpliendo con todos los objetivos planteados.

6. Líneas de futuro.

6.1 Desarrollo

Este proyecto se ha focalizado en procesos de transformación que entran en la definición de *ETL* y en cómo realizarlo de forma estable en un entorno que se podría considerar como Big Data, dado que el uso corriente de las herramientas habituales no es suficiente para realizar la tarea.

Como línea de futuro de mejora del sistema actual, se encuentra la refinación de atributos utilizados para mejorar la utilidad de las visualizaciones actuales, y el despliegue del código *ETL* en un sistema cloud.

Como línea de futuro de ampliación del sistema actual, se encuentra generar nuevas capas de conocimiento que no sean extraíbles desde el motor de visualización (por ejemplo, las nubes de palabras, la clusterización o la predicción lineal incluso con series temporales son realizables desde Tableau) junto con la realización de nuevos objetivos de visualización.

6.1 Vida y estratégica del proyecto

Ahora se podría considerar que el proyecto se encuentra en fase alfa, y es una demo.
Dado que el coste de mantenimiento del proyecto es mínimo y puede llegar a no requerir ningún elemento persistente en un entorno local, o sería de coste mínimo, es factible la continuación del proyecto como trabajo en segundo plano en cuanto a recursos. Dado este escenario, a priori el proyecto toma una hoja de ruta hacia el concepto de datos abiertos, en la que se incluirán las redacciones de contenido para la página, el renombrado y la adquisición de un dominio web, la adquisición de un hosting web compartido de bajo coste, la recopilación de necesidades de potenciales usuarios, la adaptación del contenido, etc. El objetivo a corto plazo es la puesta a punto de un sistema aceptable para el público(Q1 2020). El objetivo a medio plazo es encontrar y obtener feedback de perfiles de usuario reales para la mejora y adaptación del sistema (Q1 2021) y el objetivo a largo plazo es la entrada y asentamiento en un nicho del mercado(Q1 2022). Una vez obtenido reconocimiento, será posible buscar colaboraciones con entidades interesadas.

Bibliografia

[1] Clinicaltrials.gov. (2019). *Trends, Charts, and Maps - ClinicalTrials.gov*. [online] Available at: https://clinicaltrials.gov/ct2/resources/trends [Accessed 8 Jun. 2019].

[2] Phrma-docs.phrma.org. (2019). *Biopharmaceutical Industry-Sponsored Clinical Trials: Impact on State Economies.*. [online] Available at: http://phrma-docs.phrma.org/sites/default/files/pdf/biopharmaceutical-industry-sponsored-clinical-trials-impact-on-state-economies.pdf [Accessed 8 Jun. 2019].

[3] Clinicaltrials.gov. (2019). *ClinicalTrials.gov Background - ClinicalTrials.gov*. [online] Available at: https://clinicaltrials.gov/ct2/about-site/background [Accessed 8 Jun. 2019].

[4] raywenderlich.com. (2019). *Scrum Of One: How to Bring Scrum into your One-Person Operation*. [online] Available at: https://www.raywenderlich.com/585-scrum-of-one-how-to-bring-scrum-into-your-one-person-operation [Accessed 8 Jun. 2019].

[5] Ecured.cu. (2019). *Modelo de prototipos - EcuRed*. [online] Available at: https://www.ecured.cu/Modelo_de_prototipos [Accessed 8 Jun. 2019].

[6] Pentaho Documentation. (2019). *Learn About the PDI Client (Spoon)*. [online] Available at: https://help.pentaho.com/Documentation/8.1/Products/Data_Integration/PDI_Client [Accessed 9 Jun. 2019].

[7] Google.es. (2019). *Hojas de cálculo de Google: crea y edita hojas de cálculo online de forma gratuita.*. [online] Available at: https://www.google.es/intl/es/sheets/about/ [Accessed 8 Jun. 2019].

[8] Tableau Public. (2019). *Tableau Public*. [online] Available at: https://public.tableau.com/s/ [Accessed 8 Jun. 2019].

[9] Kaggle.com. (2019). *Kaggle: Your Home for Data Science*. [online] Available at: https://www.kaggle.com/ [Accessed 9 Jun. 2019].

[10] Pandas.pydata.org. (2019). *Python Data Analysis Library — pandas: Python Data Analysis Library*. [online] Available at: https://pandas.pydata.org/ [Accessed 8 Jun. 2019].

[11] Jupyter.org. (2019). *Project Jupyter*. [online] Available at: https://jupyter.org/ [Accessed 8 Jun. 2019].

[12] Anaconda. (2019). *Home - Anaconda*. [online] Available at: https://www.anaconda.com/ [Accessed 8 Jun. 2019].

[13] Databricks. (2019). *Databricks Community Edition FAQs*. [online] Available at: https://databricks.com/product/faq/community-edition [Accessed 8 Jun. 2019].

[14] Google.com. (2019). *Google Drive: almacenamiento en la nube, copias de seguridad de fotos, documentos y mucho más*. [online] Available at: https://www.google.com/intl/es_ALL/drive/ [Accessed 8 Jun. 2019].

[15] Tableau Software. (2019). *Tableau Desktop*. [online] Available at: https://www.tableau.com/es-es/products/desktop [Accessed 8 Jun. 2019].

[16] Es.wikipedia.org. (2019). *WYSIWYG*. [online] Available at: https://es.wikipedia.org/wiki/WYSIWYG [Accessed 8 Jun. 2019].

[17] Amazon Web Services, Inc. (2019). *Capa gratuita de Amazon RDS – Amazon Web Services (AWS)*. [online] Available at: https://aws.amazon.com/es/rds/free/ [Accessed 8 Jun. 2019].

[18] Hacker Noon. (2019). *Has the Python GIL been slain?*. [online] Available at: https://hackernoon.com/has-the-python-gil-been-slain-9440d28fa93d [Accessed 8 Jun. 2019].

[19] Es.wikipedia.org. (2019). *Desarrollo guiado por pruebas*. [online] Available at: https://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas [Accessed 8 Jun. 2019].

Apéndice 1 - Código python ETL

Este código en python del proceso *ETL* incluye los textos resultantes su ejecución. Al estar realizado en un *workbook*, cada cuadro de *input* está marcado y numerado.

In [1]:

```
import urllib
from tqdm import tqdm_notebook as tqdm #
import zipfile
import os
import xmltodict
import json
import pandas as pd
from pandas.io.json import json_normalize
import shutil
import pandas as pd
import psutil
import gc
from multiprocessing import Process
import multiprocessing
import numpy as np
import ast
import pygsheets
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from flatten_json import flatten
import glob
from xlsxwriter.workbook import Workbook
import requests
```

In [2]:

```
OUT_DIR = '/parsed'
TMP_DIR = '/tmp'

#Try to Clean ALL. Some exceptions may ocurr
try:
    if os.path.exists(OUT_DIR):
        shutil.rmtree(OUT_DIR)
except:
    pass


#Setup Base Directories
if not os.path.exists(OUT_DIR):
```

```
    os.makedirs(OUT_DIR)
if not os.path.exists(TMP_DIR):
    os.makedirs(TMP_DIR)
if not os.path.exists(OUT_DIR+TMP_DIR):
    os.makedirs(OUT_DIR+TMP_DIR)

#Define constants
RELEVANT_DATA_XNAME = '/allRelevantAttributes'
RELEVANT_EXPLODED_LOCATIONS_DATA_XNAME = '/allRelevantAttributesExplodedLocation'
CONDITIONS_DATA_XNAME = '/explodedConditions'
INVESTIGATORS_DATA_XNAME= '/explodedInvestigator'
INVESTIGATORS_CONDITIONS_DATA_XNAME = '/explodedInvestigatorAndConditions'

RELEVANT_DATA_PATH = OUT_DIR+RELEVANT_DATA_XNAME
RELEVANT_EXPLODED_LOCATIONS_DATA_PATH =
OUT_DIR+RELEVANT_EXPLODED_LOCATIONS_DATA_XNAME
CONDITIONS_DATA_PATH = OUT_DIR+CONDITIONS_DATA_XNAME
INVESTIGATORS_DATA_PATH = OUT_DIR+INVESTIGATORS_DATA_XNAME
INVESTIGATORS_CONDITIONS_DATA_PATH =
OUT_DIR+INVESTIGATORS_CONDITIONS_DATA_XNAME

RELEVANT_DATA_FILE = OUT_DIR+RELEVANT_DATA_XNAME+".csv"
RELEVANT_EXPLODED_LOCATIONS_DATA_FILE
=OUT_DIR+RELEVANT_EXPLODED_LOCATIONS_DATA_XNAME+".csv"
CONDITIONS_DATA_FILE = OUT_DIR+CONDITIONS_DATA_XNAME+".csv"
INVESTIGATORS_DATA_FILE= OUT_DIR+INVESTIGATORS_DATA_XNAME+".csv"
INVESTIGATORS_CONDITIONS_DATA_FILE =
OUT_DIR+INVESTIGATORS_CONDITIONS_DATA_XNAME+".csv"

RELEVANT_DATA_FILE_EXCEL = OUT_DIR+RELEVANT_DATA_XNAME+".xlsx"
RELEVANT_EXPLODED_LOCATIONS_DATA_FILE_EXCEL
=OUT_DIR+RELEVANT_EXPLODED_LOCATIONS_DATA_XNAME+".xlsx"
CONDITIONS_DATA_FILE_EXCEL = OUT_DIR+CONDITIONS_DATA_XNAME+".xlsx"
INVESTIGATORS_DATA_FILE_EXCEL= OUT_DIR+INVESTIGATORS_DATA_XNAME+".xlsx"
INVESTIGATORS_CONDITIONS_DATA_FILE_EXCEL =
OUT_DIR+INVESTIGATORS_CONDITIONS_DATA_XNAME+".xlsx"
```

In [3]:

```
#Define attributes to use on each dataframe
full_data_attributes =["clinical_study.location","clinical_study.location.investigator",
"clinical_study.sponsors.collaborator",
"clinical_study.clinical_results.point_of_contact.email","clinical_study.clinical_results.point_of_conta
ct.name_or_title","clinical_study.clinical_results.point_of_contact.organization","clinical_study.clinic
al_results.point_of_contact.phone","clinical_study.condition","clinical_study.condition_browse.mes
```

h_term","clinical_study.eligibility.gender","clinical_study.eligibility.maximum_age","clinical_study.eligibility.minimum_age","clinical_study.id_info.nct_id","clinical_study.id_info.org_study_id","clinical_study.keyword","clinical_study.last_known_status","clinical_study.last_update_submitted","clinical_study.last_update_submitted_qc","clinical_study.location.contact.last_name","clinical_study.location.contact.phone","clinical_study.location.contact_backup.email","clinical_study.location.facility.address.city","clinical_study.location.facility.address.country","clinical_study.location.facility.address.state","clinical_study.location.facility.address.zip","clinical_study.location.facility.name","clinical_study.location.investigator.last_name","clinical_study.location.investigator.role","clinical_study.location.status","clinical_study.location_countries.country","clinical_study.official_title","clinical_study.overall_contact.email","clinical_study.overall_contact.last_name","clinical_study.overall_contact.phone","clinical_study.overall_contact_backup.email","clinical_study.overall_official.affiliation","clinical_study.overall_official.last_name","clinical_study.overall_official.role","clinical_study.overall_status","clinical_study.oversight_info.has_dmc","clinical_study.oversight_info.is_fda_regulated_device","clinical_study.oversight_info.is_fda_regulated_drug","clinical_study.oversight_info.is_ppsd","clinical_study.phase","clinical_study.responsible_party.responsible_party_type","clinical_study.results_first_submitted","clinical_study.results_first_submitted_qc","clinical_study.source","clinical_study.sponsors.collaborator.agency","clinical_study.sponsors.collaborator.agency_class","clinical_study.sponsors.lead_sponsor.agency","clinical_study.sponsors.lead_sponsor.agency_class","clinical_study.start_date","clinical_study.study_design_info.primary_purpose","clinical_study.study_first_submitted_qc","clinical_study.study_type","clinical_study.verification_date","clinical_study.why_stopped"]
relevant_data_attributes =["condition","clinical_study.sponsors.collaborator",
"clinical_study.clinical_results.point_of_contact.email","clinical_study.clinical_results.point_of_contact.name_or_title","clinical_study.clinical_results.point_of_contact.organization","clinical_study.clinical_results.point_of_contact.phone",
"clinical_study.eligibility.gender","clinical_study.eligibility.maximum_age","clinical_study.eligibility.minimum_age","clinical_study.id_info.nct_id","clinical_study.id_info.org_study_id","clinical_study.keyword","clinical_study.last_known_status","clinical_study.last_update_submitted","clinical_study.last_update_submitted_qc","clinical_study.location.contact.last_name","clinical_study.location.contact.phone","clinical_study.location.contact_backup.email","clinical_study.location.facility.address.city","clinical_study.location.facility.address.country","clinical_study.location.facility.address.state","clinical_study.location.facility.address.zip","clinical_study.location.facility.name","clinical_study.location.investigator.last_name","clinical_study.location.investigator.role","clinical_study.location.status","clinical_study.location_countries.country","clinical_study.official_title","clinical_study.overall_contact.email","clinical_study.overall_contact.last_name","clinical_study.overall_contact.phone","clinical_study.overall_contact_backup.email","clinical_study.overall_official.affiliation","clinical_study.overall_official.last_name","clinical_study.overall_official.role","clinical_study.overall_status","clinical_study.oversight_info.has_dmc","clinical_study.oversight_info.is_fda_regulated_device","clinical_study.oversight_info.is_fda_regulated_drug","clinical_study.oversight_info.is_ppsd","clinical_study.phase","clinical_study.responsible_party.responsible_party_type","clinical_study.results_first_submitted","clinical_study.results_first_submitted_qc","clinical_study.source","clinical_study.sponsors.collaborator.agency","clinical_study.sponsors.collaborator.agency_class","clinical_study.sponsors.lead_sponsor.agency","clinical_study.sponsors.lead_sponsor.agency_class","clinical_study.start_date","clinical_study.study_design_info.primary_purpose","clinical_study.study_first_submitted_qc","clinical_study.study_type","clinical_study.verification_date","clinical_study.why_stopped"]

```
explodeConditions_data_attributes =
["condition","clinical_study.id_info.nct_id","clinical_study.study_first_submitted_qc","clinical_study.last_update_submitted_qc","clinical_study.location.facility.address.city","clinical_study.location.facility.address.country","clinical_study.location.facility.address.state","clinical_study.location.facility.address.zip"]
explodeInvestigatorsPlusConditions_data_attributes = explodeConditions_data_attributes.copy()
explodeInvestigatorsPlusConditions_data_attributes.extend(["clinical_study.location.investigator","clinical_study.location.investigator.last_name","clinical_study.location.investigator.role","clinical_study.overall_contact.email","clinical_study.overall_contact.last_name","clinical_study.overall_contact.phone","clinical_study.overall_official.affiliation","clinical_study.overall_official.last_name","clinical_study.overall_official.role"])
```

In [4]:

```
#Login to Gooogle Drive (Human intervention required.)
#Compatible with jupyter nbconvert
gauth = GoogleAuth()
# Try to load saved client credentials
gauth.LoadCredentialsFile("/tmp/credentials/mycreds.txt")
if gauth.credentials is None:
    # Authenticate if they're not there
    gauth.LocalWebserverAuth()
elif gauth.access_token_expired:
    # Refresh them if expired
    gauth.Refresh()
else:
    # Initialize the saved creds
    gauth.Authorize()

# Save the current credentials to a file
gauth.SaveCredentialsFile("/tmp/credentials/mycreds.txt")

drive = GoogleDrive(gauth)
```

In [5]:

```
#Global Helper Functions
def cleanConditions(row):
    data = []
    r1= row['clinical_study.condition']
    r2 = row['clinical_study.condition_browse.mesh_term']
    if isinstance(r1, str):
        data.append(r1)
    else:
        data.extend(r1)
    if isinstance(r2, str):
```

```
        data.append(r2)
    else:
        data.extend(r2)
    return list(set(data))

def downloadChunk(url, chunkNum):
    global haveMoreChunks
    strChunkNum = str(chunkNum)

    # NOTE the stream=True parameter below
    with requests.get(url, stream=True) as r:
        if r.status_code< 400:
            with open(TMP_DIR+"/chunk"+strChunkNum+".zip", 'wb') as f:
                for chunk in r.iter_content(chunk_size=8192):
                    if chunk: # filter out keep-alive new chunks
                        f.write(chunk)
                        f.flush()
        else:
            haveMoreChunks=False

def explode(df, lst_cols, fill_value='', preserve_index=False):
    # make sure `lst_cols` is list-alike
    if (lst_cols is not None
        and len(lst_cols) > 0
        and not isinstance(lst_cols, (list, tuple, np.ndarray, pd.Series))):
        lst_cols = [lst_cols]
    # all columns except `lst_cols`
    idx_cols = df.columns.difference(lst_cols)
    # calculate lengths of lists
    lens = df[lst_cols[0]].str.len()
    # preserve original index values
    idx = np.repeat(df.index.values, lens)
    # create "exploded" DF
    res = (pd.DataFrame({
                col:np.repeat(df[col].values, lens)
                for col in idx_cols},
                index=idx)
            .assign(**{col:np.concatenate(df.loc[lens>0, col].values)
                        for col in lst_cols}))
    # append those rows that have empty lists
    if (lens == 0).any():
        # at least one list in cells is empty
        res = (res.append(df.loc[lens==0, idx_cols], sort=False)
                .fillna(fill_value))
    # revert the original index order
```

```
    res = res.sort_index()
    # reset index if requested
    if not preserve_index:
        res = res.reset_index(drop=True)
    return res

def recursivelyExpandNestedJson(df, i, path, node):
    thisLevelPath=path
    for key in node.keys():
        path = thisLevelPath+"."+key
        value = node.get(key)
        if isinstance(value, dict):
            recursivelyExpandNestedJson(df, i, path, value)
        elif path in df.columns:
            df.at[i, path] = value

def reduce_mem_usage(df, verbose=True):
    """
    Test pending. Not using this yet.
    """
    numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']
    start_mem = df.memory_usage().sum() / 1024**2
    for col in df.columns:
        col_type = df[col].dtypes
        if col_type in numerics:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max < np.finfo(np.float16).max:
                    df[col] = df[col].astype(np.float16)
                elif c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).max:
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)
    end_mem = df.memory_usage().sum() / 1024**2
```

```
    if verbose: print('Mem. usage decreased to {:5.2f} Mb ({:.1f}% reduction)'.format(end_mem, 100 *
(start_mem - end_mem) / start_mem))
    return df
```

In [6]:

```
def readFile(file):
    """
    Helper function to read a file.
    """
    with open(file,'r', encoding="utf8") as f:
        return f.read()

def processChunk(strChunkNum):
    """
    Reads downloaded data chunk by id, process data and append it to files.
    """
    print("Processing chunk "+strChunkNum)

    #Create a list of files inside chunk
    filelist=[]
    for folder, subs, files in os.walk(TMP_DIR+"/unzippedChunk"+strChunkNum):
        for filename in files:
            if filename.endswith(".xml"):
                filelist.append(os.path.join(folder, filename))
    #Loads one dataframe for each file, filtering desired rows.
    print("Loading files into Dataframe...")
    df_list = [json_normalize(xmltodict.parse(readFile(file))).filter(full_data_attributes, axis=1) for file in
tqdm(filelist, mininterval =1.0)]
    #Create one big dataframe from small ones.
    big_df = pd.concat(df_list, sort=True)
    big_df.reset_index()

    print("Cleaning Dataframe.")
    #General Cleans on df
    big_df=big_df.fillna("")
    big_df['condition'] = big_df.apply(lambda row: cleanConditions(row),axis=1)
    big_df.drop(['clinical_study.condition','clinical_study.condition_browse.mesh_term'], axis=1,
inplace=True)#Optional

    #Expand location for cases of multiple locations on the same clinical trial
    big_df_explodedLocations= explode(big_df, ['clinical_study.location'], fill_value='',
preserve_index=False)
    for i, row in big_df_explodedLocations.iterrows():
        location = row['clinical_study.location']
```

```
        if isinstance(location, dict):
            path = 'clinical_study.location'
            recursivelyExpandNestedJson(big_df_explodedLocations, i, path, location)
    big_df.drop(['clinical_study.location'], axis=1, inplace=True)#Optional

    print("Processing Dataframe variations and storing data into target files...")

    #Single thread core version. Multicore didn't work as expected.

    #Prepare Dataframe into memory
    relevantDataAttributes_df = big_df.filter(relevant_data_attributes, axis=1)
    #Clean unused source Dataframe from memory
    del big_df
    gc.collect()
    #Process and store Dataframe
    processRelevantDataAttributesDataframe(relevantDataAttributes_df, strChunkNum)
    #Clean Dataframe From memory
    del relevantDataAttributes_df
    gc.collect()

    #Prepare Dataframe into memory
    relevantDataAttributes_explodedLocations_df =
big_df_explodedLocations.filter(relevant_data_attributes, axis=1)
    #Process and store Dataframe

processRelevantDataAttributesDataframeExplodedLocations(relevantDataAttributes_explodedLocat
ions_df, strChunkNum)
    #Clean Dataframe From memory
    del relevantDataAttributes_explodedLocations_df
    gc.collect()

    #Prepare Dataframe into memory
    explodeConditions_df = big_df_explodedLocations.filter(explodeConditions_data_attributes,
axis=1) #use exploded locations
    #Process and store Dataframe
    processExplodeConditionsDataframe(explodeConditions_df, strChunkNum)
    #Clean Dataframe From memory
    del explodeConditions_df
    gc.collect()

    #Prepare Dataframe into memory
    explodeInvestigatorsAndConditions_df =
big_df_explodedLocations.filter(explodeInvestigatorsPlusConditions_data_attributes, axis=1) #use
exploded locations
    #Process and store Dataframe
```

```
    processExplodeInvestigatorsDataframe(explodeInvestigatorsAndConditions_df, strChunkNum)
    #Clean Dataframe From memory
    del explodeInvestigatorsAndConditions_df

    #Finish
    del big_df_explodedLocations
    gc.collect()
    print(psutil.virtual_memory())


def processRelevantDataAttributesDataframe(df, strChunkNum):
    global relevantRow
    global relevantWorksheet
    global relevantWorkBook
    global maxRowsPerSheet

    if relevantRow >= maxRowsPerSheet:
        relevantWorksheet=relevantWorkBook.add_worksheet()
        relevantRow=0


    templateIndexes = []
    ncols= len(df.columns)

    #Setup column indexes
    for column in df.columns:
        index = relevant_data_attributes.index(column)
        templateIndexes.append(index)

    #If new file vWrite column header
    if(relevantRow==0):
        for ind, colName in enumerate(relevant_data_attributes):
            relevantWorksheet.write(0, ind, colName)
        relevantRow = 1
    realR =0
    for r, row in enumerate(df.itertuples(index=False)):
        realR = r+relevantRow
        for c in range(0, ncols):
            cell = row[c]
            if isinstance(cell, list):
                cell = str(cell)
            relevantWorksheet.write(realR, templateIndexes[c], cell)
    relevantRow = realR

def processRelevantDataAttributesDataframeExplodedLocations(df, strChunkNum):
```

```
    global relevantExplodedRow
    global relevantExplodedWorksheet
    global relevantExplodedWorkbook
    global maxRowsPerSheet

    if relevantExplodedRow >= maxRowsPerSheet:
        relevantExplodedWorksheet=relevantExplodedWorkbook.add_worksheet()
        relevantExplodedRow=0

    templateIndexes = []
    ncols= len(df.columns)

    #Setup column indexes
    for column in df.columns:
        index = relevant_data_attributes.index(column)
        templateIndexes.append(index)

    #If new file vWrite column header
    if(relevantExplodedRow==0):
        for ind, colName in enumerate(relevant_data_attributes):
            relevantExplodedWorksheet.write(0, ind, colName)
        relevantExplodedRow = 1

    realR =0
    for r, row in enumerate(df.itertuples(index=False)):
        realR = r+relevantExplodedRow
        for c in range(0, ncols):
            cell = row[c]
            if isinstance(cell, list):
                cell = str(cell)
            relevantExplodedWorksheet.write(realR, templateIndexes[c], cell)
    relevantExplodedRow = realR

def processExplodeConditionsDataframe(df, strChunkNum):
    global conditionsRow
    global conditionsWorksheet
    global conditionsWorkbook
    global maxRowsPerSheet

    if conditionsRow >= maxRowsPerSheet:
        conditionsWorksheet=conditionsWorkbook.add_worksheet()
        conditionsRow=0

    #Explode Conditions
    df= explode(df, ['condition'], fill_value='', preserve_index=False)
```

```
    templateIndexes = []
    ncols= len(df.columns)

    #Setup column indexes
    for column in df.columns:
        index = explodeConditions_data_attributes.index(column)
        templateIndexes.append(index)

    #If new file vWrite column header
    if(conditionsRow==0):
        for ind, colName in enumerate(explodeConditions_data_attributes):
            conditionsWorksheet.write(0, ind, colName)
        conditionsRow = 1

    realR =0
    for r, row in enumerate(df.itertuples(index=False)):
        realR = r+conditionsRow
        for c in range(0, ncols):
            cell = row[c]
            if isinstance(cell, list):
                cell = str(cell)
            conditionsWorksheet.write(realR, templateIndexes[c], cell)
    conditionsRow = realR

def processExplodeInvestigatorsDataframe(df, strChunkNum):
    global investigatorsRow
    global investigatorsConditionsRow

    global investigatorsWorksheet
    global investigatorsConditionsWorksheet

    global investigatorsWorkbook
    global investigatorsConditionsWorkbook

    global maxRowsPerSheet

    if investigatorsRow >= maxRowsPerSheet:
        investigatorsWorksheet=investigatorsWorkbook.add_worksheet()
        investigatorsRow=0
    if investigatorsConditionsRow >= maxRowsPerSheet:
        investigatorsConditionsWorksheet=investigatorsConditionsWorkbook.add_worksheet()
        investigatorsConditionsRow=0

    #Explode investigators
```

```
    try:
        df= explode(df, ['clinical_study.location.investigator'], fill_value='', preserve_index=False)
        for i, row in df.iterrows():
            if isinstance(row['clinical_study.location.investigator'], dict):
                df.at[i,'clinical_study.location.investigator.last_name'] =
row['clinical_study.location.investigator'].get('last_name')
                df.at[i,'clinical_study.location.investigator.role'] =
row['clinical_study.location.investigator'].get('role')
        df.drop("clinical_study.location.investigator", axis=1, inplace=True)
    except:
        pass

    templateIndexes = []
    ncols= len(df.columns)

    #If new file vWrite column header
    for column in df.columns:
        index = explodeInvestigatorsPlusConditions_data_attributes.index(column)
        templateIndexes.append(index)

    #Write column header
    if(investigatorsRow==0):
        for ind, colName in enumerate(explodeInvestigatorsPlusConditions_data_attributes):
            investigatorsWorksheet.write(0, ind, colName)
        investigatorsRow = 1

    realR =0
    for r, row in enumerate(df.itertuples(index=False)):
        realR = r+investigatorsRow
        for c in range(0, ncols):
            cell = row[c]
            if isinstance(cell, list):
                cell = str(cell)
            investigatorsWorksheet.write(realR, templateIndexes[c], cell)
    investigatorsRow = realR

    #Explode conditions
    df= explode(df, ['condition'], fill_value='', preserve_index=False)

    templateIndexes = []
    ncols= len(df.columns)

    #Setup column indexes
    for column in df.columns:
        index = explodeInvestigatorsPlusConditions_data_attributes.index(column)
```

```
        templateIndexes.append(index)

    #If new file vWrite column header
    if(investigatorsConditionsRow==0):
        for ind, colName in enumerate(explodeInvestigatorsPlusConditions_data_attributes):
            investigatorsConditionsWorksheet.write(0, ind, colName)
        investigatorsConditionsRow = 1

    realR =0
    for r, row in enumerate(df.itertuples(index=False)):
        realR = r+investigatorsConditionsRow
        for c in range(0, ncols):
            cell = row[c]
            if isinstance(cell, list):
                cell = str(cell)
            investigatorsConditionsWorksheet.write(realR, templateIndexes[c], cell)
    investigatorsConditionsRow = realR
```

In [7]:

```
def publishToGoogleDrive(drive, DATA_FILE):
    """
    Publish data to Google drive. Adhoc filename operation to remove slash
    """
    file_list = drive.ListFile({'q': "title='"+DATA_FILE[1:]+"' and trashed=false"}).GetList()
    for file in file_list:
        print(file['title'], file['id'])
        file.SetContentFile(OUT_DIR+DATA_FILE)
        file.Upload()
```

In [8]:

```
#Define and open Global writers for final finles
global relevantRow
global relevantExplodedRow
global conditionsRow
global investigatorsRow
global investigatorsConditionsRow

global maxRowsPerSheet
maxRowsPerSheet =  = 900000

relevantRow=0
relevantExplodedRow=0
conditionsRow = 0
investigatorsRow = 0
```

```
investigatorsConditionsRow = 0

global relevantWorkbook
global relevantWorksheet
global relevantExplodedWorkbook
global relevantExplodedWorksheet
global conditionsWorkbook
global conditionsWorksheet
global investigatorsWorkbook
global investigatorsWorksheet
global investigatorsConditionsWorkbook
global investigatorsConditionsWorksheet

relevantWorkbook = Workbook(RELEVANT_DATA_FILE_EXCEL, {'constant_memory':
True,'strings_to_numbers':True, 'tmpdir': OUT_DIR+TMP_DIR})
relevantWorkbook.use_zip64()
relevantWorksheet = relevantWorkbook.add_worksheet()

relevantExplodedWorkbook =
Workbook(RELEVANT_EXPLODED_LOCATIONS_DATA_FILE_EXCEL, {'constant_memory':
True,'strings_to_numbers':True, 'tmpdir': OUT_DIR+TMP_DIR})
relevantExplodedWorkbook.use_zip64()
relevantExplodedWorksheet = relevantExplodedWorkbook.add_worksheet()

conditionsWorkbook = Workbook(CONDITIONS_DATA_FILE_EXCEL, {'constant_memory':
True,'strings_to_numbers':True, 'tmpdir': OUT_DIR+TMP_DIR})
conditionsWorkbook.use_zip64()
conditionsWorksheet = conditionsWorkbook.add_worksheet()

investigatorsWorkbook = Workbook(INVESTIGATORS_DATA_FILE_EXCEL, {'constant_memory':
True,'strings_to_numbers':True, 'tmpdir': OUT_DIR+TMP_DIR})
investigatorsWorkbook.use_zip64()
investigatorsWorksheet = investigatorsWorkbook.add_worksheet()

investigatorsConditionsWorkbook =
Workbook(INVESTIGATORS_CONDITIONS_DATA_FILE_EXCEL, {'constant_memory':
True,'strings_to_numbers':True, 'tmpdir': OUT_DIR+TMP_DIR})
investigatorsConditionsWorkbook.use_zip64()
investigatorsConditionsWorksheet = investigatorsConditionsWorkbook.add_worksheet()
```

In [9]:

```
%%time
# Main bucle. Download, unzip and process&write excel files for each result chunk.
```

```
ps = pygsheets.authorize(service_file='/tmp/credentials/key2.json')

#Setup
global haveMoreChunks

haveMoreChunks = True
chunkNum = 0
baseUrl = "https://clinicaltrials.gov/ct2/download_studies?&down_chunk="
strChunkNum = str(chunkNum)
url =baseUrl+strChunkNum

#Init
print("Downloading chunk "+strChunkNum)
urllib.request.urlretrieve(url, "/tmp/chunk"+strChunkNum+".zip")


#Bucle, with already 1 downloaded chunk.
while haveMoreChunks:
    alreadyStrChunkNum = str(chunkNum)

    #download new chunk
    chunkNum+=1
    strChunkNum = str(chunkNum)
    url =baseUrl+strChunkNum
    p = Process(target=downloadChunk(url, chunkNum))
    p.start()
    print("Downloading chunk "+strChunkNum)

    #unzip Already downloaded Chunk
    #Override folder
    if os.path.exists(TMP_DIR+"/unzippedChunk"+alreadyStrChunkNum):
        shutil.rmtree(TMP_DIR+"/unzippedChunk"+alreadyStrChunkNum, ignore_errors=True)
    os.mkdir(TMP_DIR+"/unzippedChunk"+alreadyStrChunkNum)

    zippedChunk = zipfile.ZipFile(TMP_DIR+"/chunk"+alreadyStrChunkNum+".zip")
    zippedChunk.extractall(TMP_DIR+"/unzippedChunk"+alreadyStrChunkNum)
    zippedChunk.close()

    #process data
    processChunk(alreadyStrChunkNum)

    #Wait for new chunk to be downloaded.
    p.join()
    gc.collect()
```

```
Downloading chunk 0
Downloading chunk 1
Processing chunk 0
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15459549184, percent=39.9, used=10244935680,
free=15459549184)
Downloading chunk 2
Processing chunk 1
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15485206528, percent=39.8, used=10219278336,
free=15485206528)
Downloading chunk 3
Processing chunk 2
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14873686016, percent=42.1, used=10830798848,
free=14873686016)
Downloading chunk 4
Processing chunk 3
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14784184320, percent=42.5, used=10920300544,
free=14784184320)
Downloading chunk 5
Processing chunk 4
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14651797504, percent=43.0, used=11052687360,
free=14651797504)
Downloading chunk 6
Processing chunk 5
```

```
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14630019072, percent=43.1, used=11074465792,
free=14630019072)
Downloading chunk 7
Processing chunk 6
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14569545728, percent=43.3, used=11134939136,
free=14569545728)
Downloading chunk 8
Processing chunk 7
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14467309568, percent=43.7, used=11237175296,
free=14467309568)
Downloading chunk 9
Processing chunk 8
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14443708416, percent=43.8, used=11260776448,
free=14443708416)
Downloading chunk 10
Processing chunk 9
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14459420672, percent=43.7, used=11245064192,
free=14459420672)
Downloading chunk 11
Processing chunk 10
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
```

```
svmem(total=25704484864, available=14408245248, percent=43.9, used=11296239616,
free=14408245248)
Downloading chunk 12
Processing chunk 11
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14344286208, percent=44.2, used=11360198656,
free=14344286208)
Downloading chunk 13
Processing chunk 12
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14414139392, percent=43.9, used=11290345472,
free=14414139392)
Downloading chunk 14
Processing chunk 13
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14405169152, percent=44.0, used=11299315712,
free=14405169152)
Downloading chunk 15
Processing chunk 14
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14500634624, percent=43.6, used=11203850240,
free=14500634624)
Downloading chunk 16
Processing chunk 15
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14547922944, percent=43.4, used=11156561920,
free=14547922944)
Downloading chunk 17
Processing chunk 16
Loading files into Dataframe...
```

```
Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14531596288, percent=43.5, used=11172888576,
free=14531596288)
Downloading chunk 18
Processing chunk 17
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14602407936, percent=43.2, used=11102076928,
free=14602407936)
Downloading chunk 19
Processing chunk 18
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14939205632, percent=41.9, used=10765279232,
free=14939205632)
Downloading chunk 20
Processing chunk 19
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14920691712, percent=42.0, used=10783793152,
free=14920691712)
Downloading chunk 21
Processing chunk 20
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14975397888, percent=41.7, used=10729086976,
free=14975397888)
Downloading chunk 22
Processing chunk 21
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15138926592, percent=41.1, used=10565558272,
free=15138926592)
```

```
Downloading chunk 23
Processing chunk 22
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15163031552, percent=41.0, used=10541453312,
free=15163031552)
Downloading chunk 24
Processing chunk 23
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15138877440, percent=41.1, used=10565607424,
free=15138877440)
Downloading chunk 25
Processing chunk 24
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15099052032, percent=41.3, used=10605432832,
free=15099052032)
Downloading chunk 26
Processing chunk 25
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15024254976, percent=41.6, used=10680229888,
free=15024254976)
Downloading chunk 27
Processing chunk 26
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15055060992, percent=41.4, used=10649423872,
free=15055060992)
Downloading chunk 28
Processing chunk 27
Loading files into Dataframe...

Cleaning Dataframe.
```

```
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15030321152, percent=41.5, used=10674163712,
free=15030321152)
Downloading chunk 29
Processing chunk 28
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15037829120, percent=41.5, used=10666655744,
free=15037829120)
Downloading chunk 30
Processing chunk 29
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15023734784, percent=41.6, used=10680750080,
free=15023734784)
Downloading chunk 31
Processing chunk 30
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=14830227456, percent=42.3, used=10874257408,
free=14830227456)
Downloading chunk 32
Processing chunk 31
Loading files into Dataframe...

Cleaning Dataframe.
Processing Dataframe variations and storing data into target files...
svmem(total=25704484864, available=15079227392, percent=41.3, used=10625257472,
free=15079227392)
Wall time: 2h 54min 47s
```

In [10]:

```
global relevantWorkbook
global relevantExplodedWorkbook
global conditionsWorkbook
global investigatorsWorkbook
global investigatorsConditionsWorkbook

relevantWorkbook.close()
```

```
relevantExplodedWorkbook.close()
conditionsWorkbook.close()
investigatorsWorkbook.close()
investigatorsConditionsWorkbook.close()
```

In [11]:

```
#Publish Data to Google Drive
publishToGoogleDrive(drive, RELEVANT_DATA_XNAME+".xlsx")
publishToGoogleDrive(drive, RELEVANT_EXPLODED_LOCATIONS_DATA_XNAME+".xlsx")
publishToGoogleDrive(drive, CONDITIONS_DATA_XNAME+".xlsx")
publishToGoogleDrive(drive, INVESTIGATORS_DATA_XNAME+".xlsx")
publishToGoogleDrive(drive, INVESTIGATORS_CONDITIONS_DATA_XNAME+".xlsx")
```

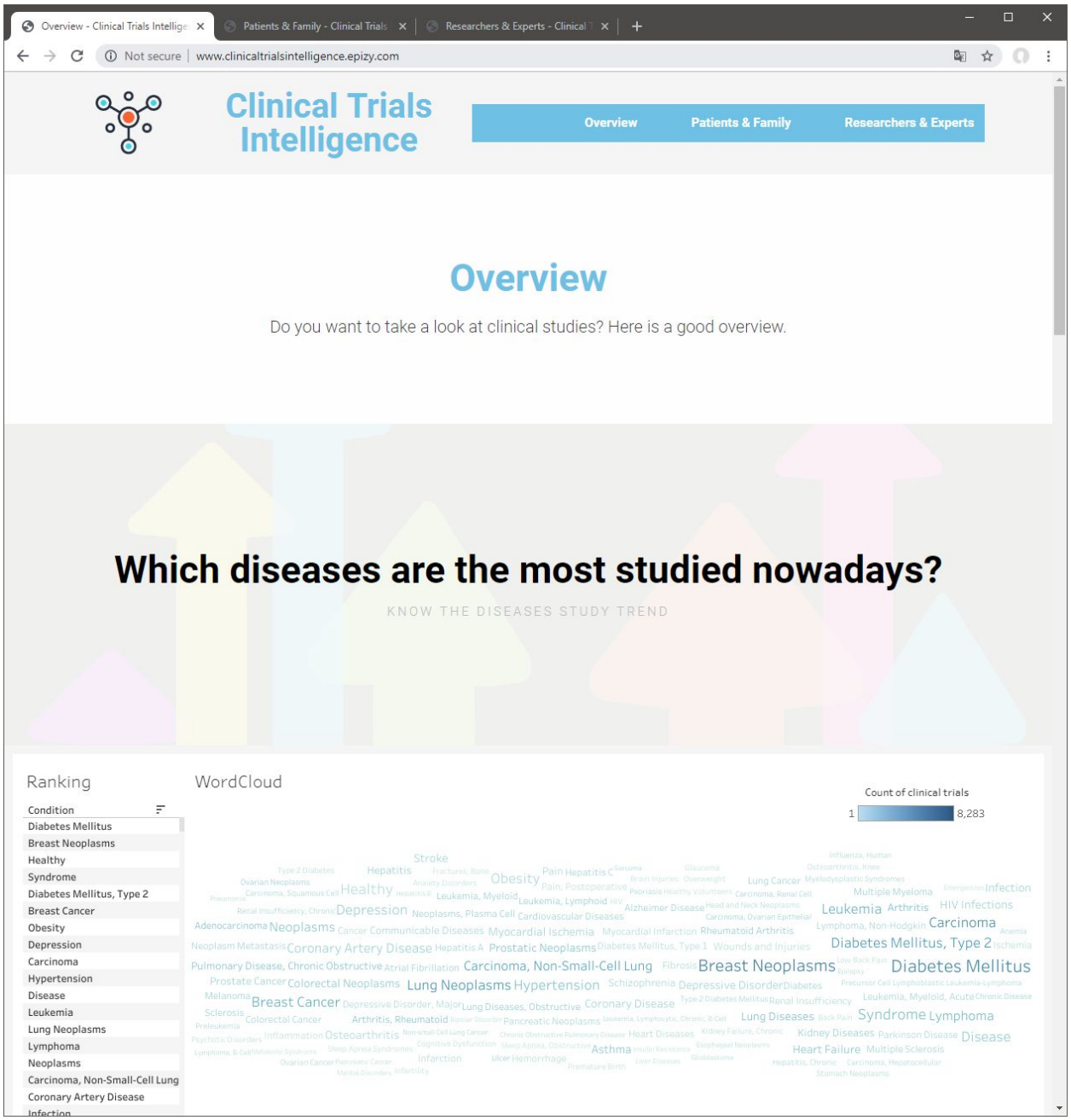
```
allRelevantAttributes.xlsx 1N-y3zaTaQrE2l6V0Q5BioPJb7whTFA8k
allRelevantAttributesExplodedLocation.xlsx 19xRsgs8rmspwUox5lYdYjFw4bEw9-T-X
explodedConditions.xlsx 1ErlO_pwPypHTvNu_PpnuHgP6y7_EBbdW
explodedInvestigator.xlsx 1Hfg5Rcowqe-JdmFVcduxb-nB_BRW5nMD
explodedInvestigatorAndConditions.xlsx 1tF4LgzlrYLJFdjM8Gdn_RfYpLAkpMmos
```

Apéndice 2 - Imágenes de la web del proyecto

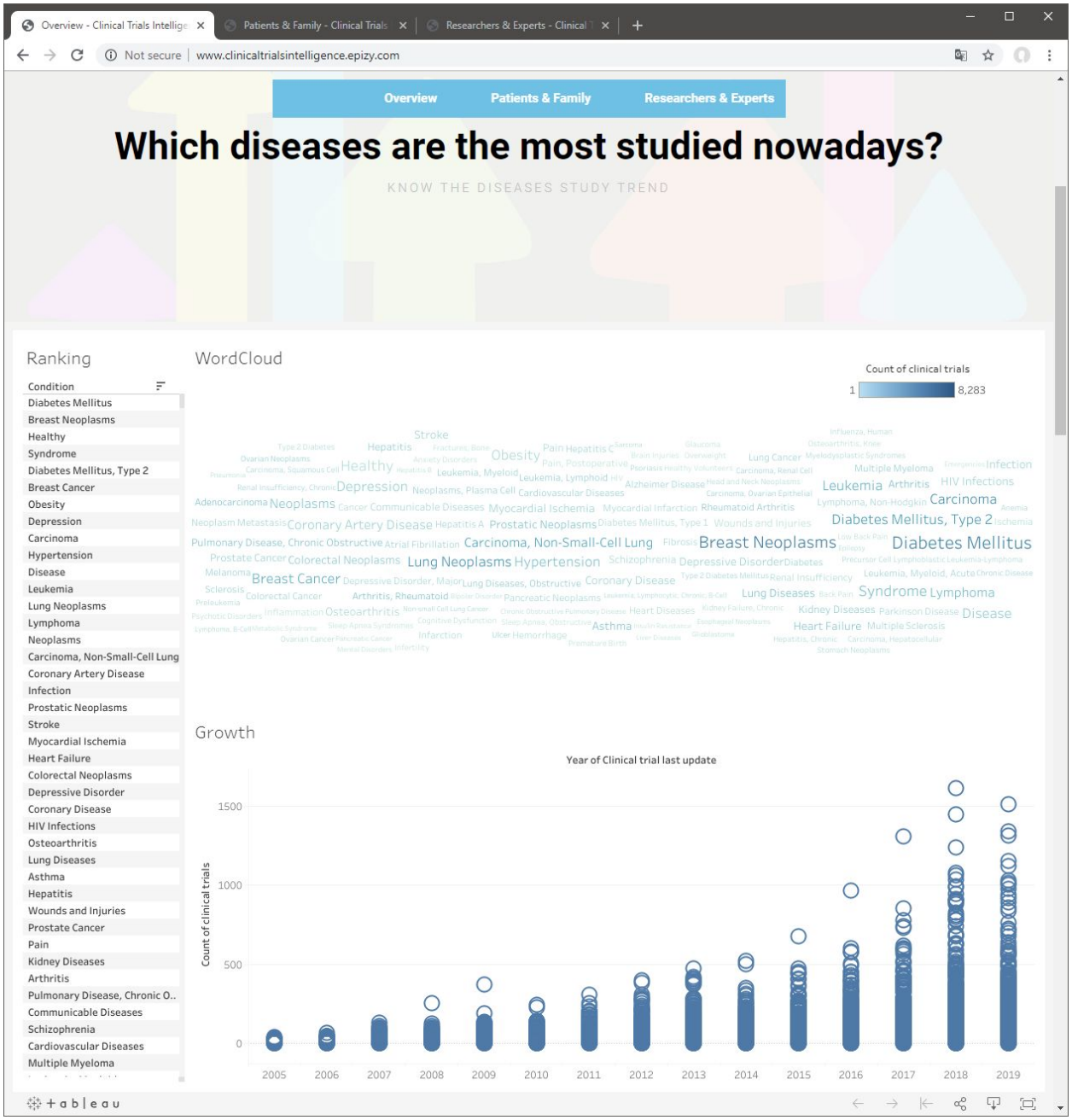
Nota: Tanto la web como la gráfica se adaptan a la pantalla, es decir, son responsive.

Apéndice 2.1 - Vistas adaptadas al tamaño del documento

Overview

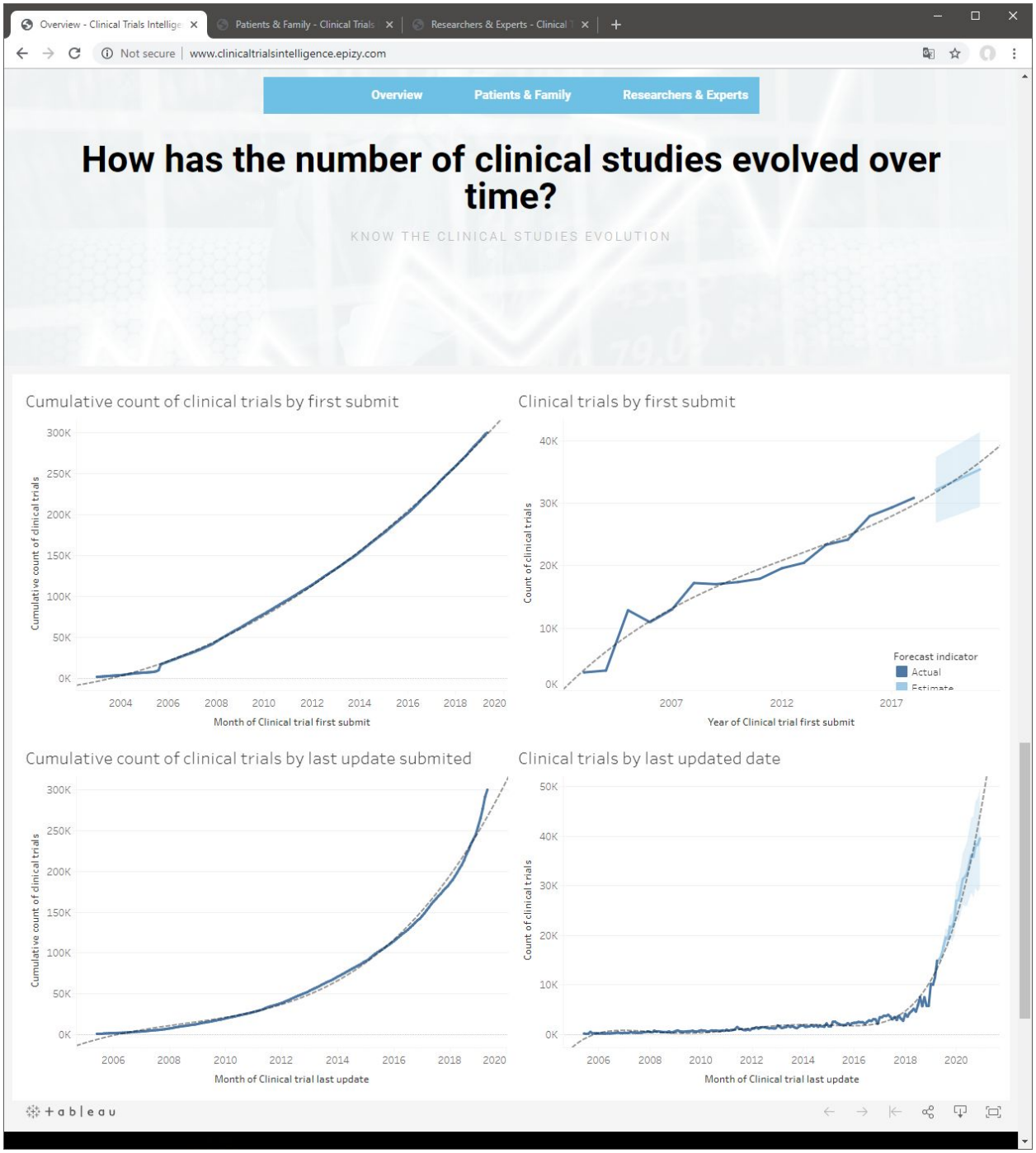


Overview



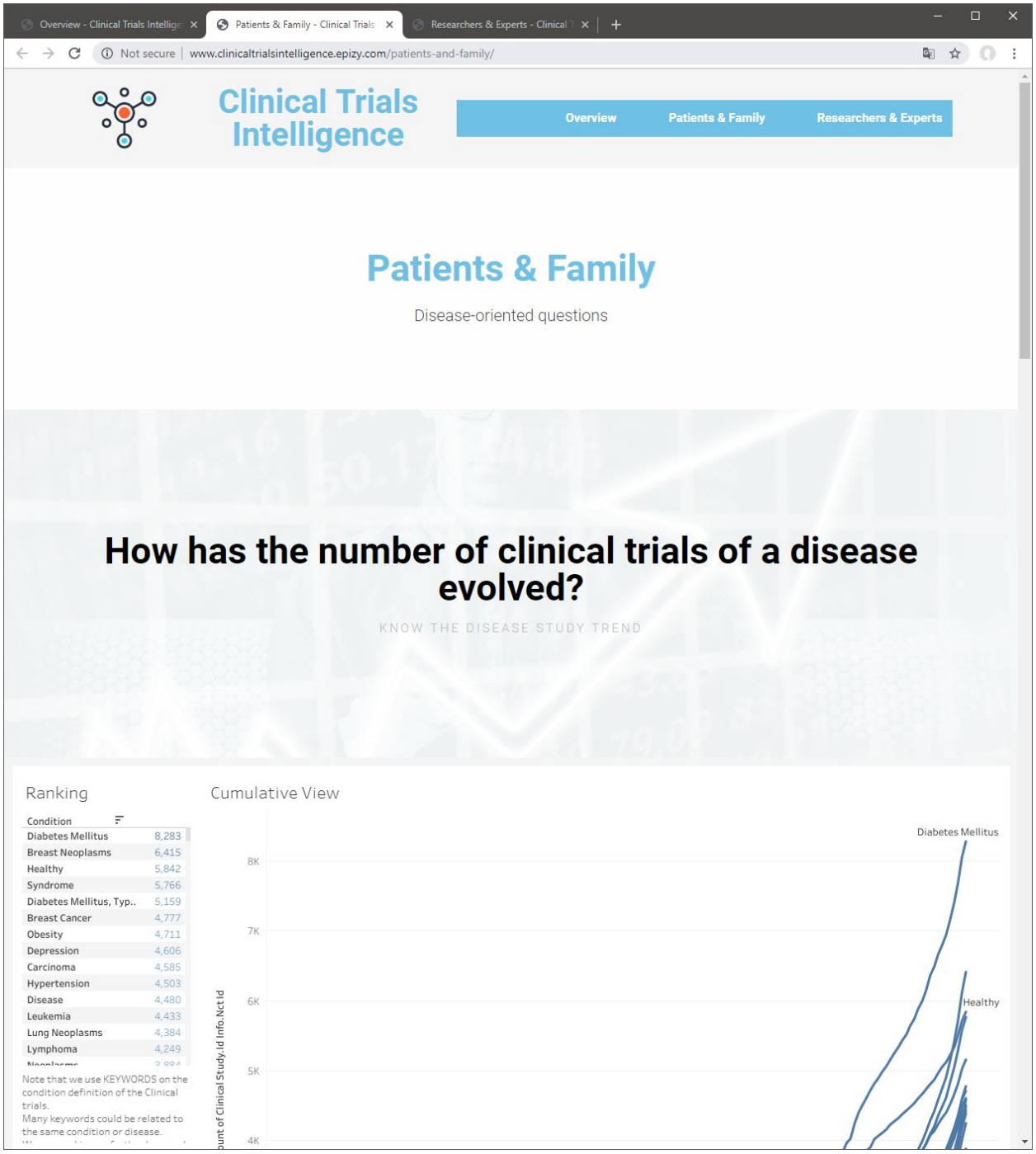
Overview - q1_studiedDiseasesNow

Overview - Q2_countriesStudiesInProgress

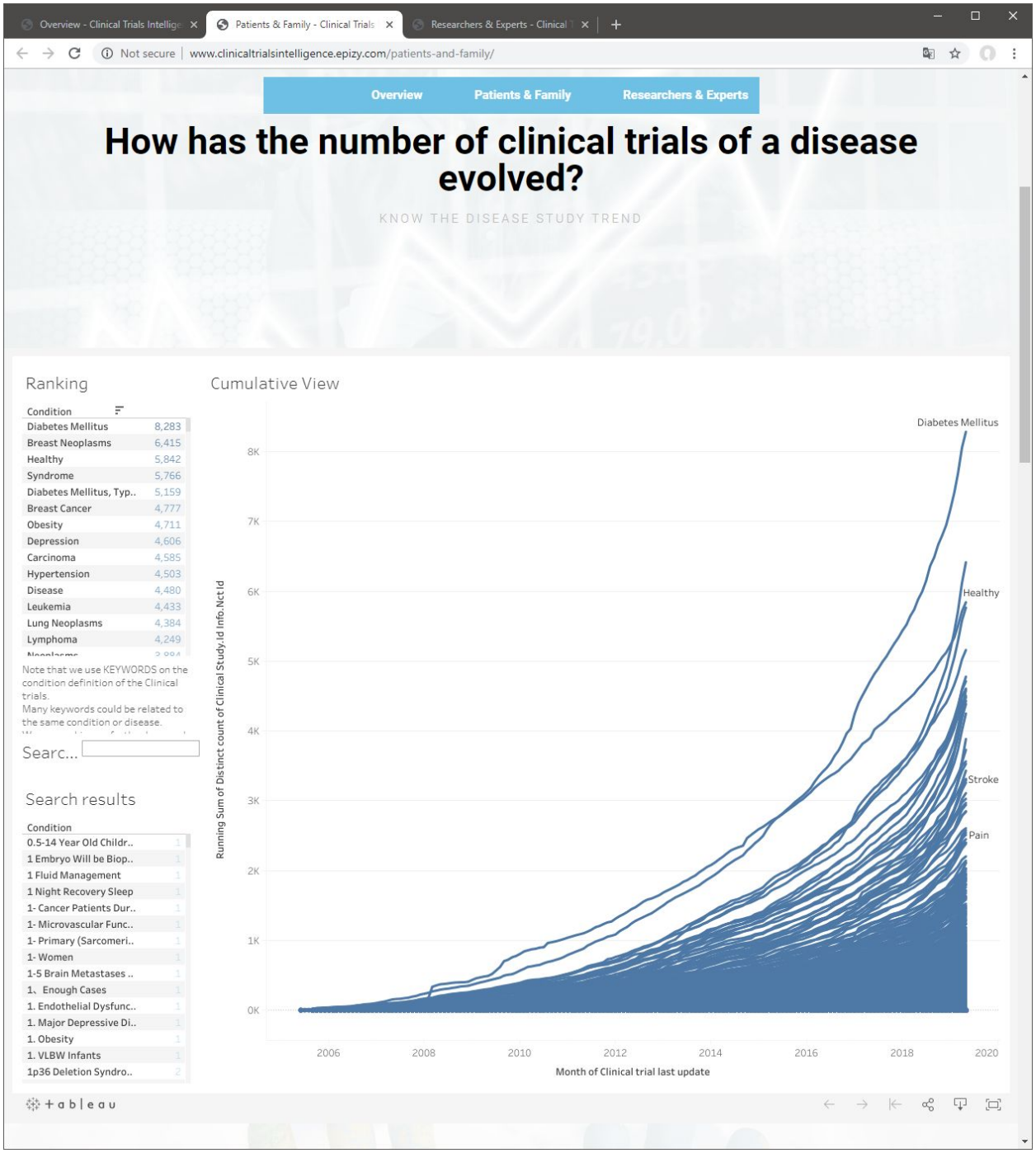


Overview - *Q3_evolutionNumberTrials*

Patients & Family

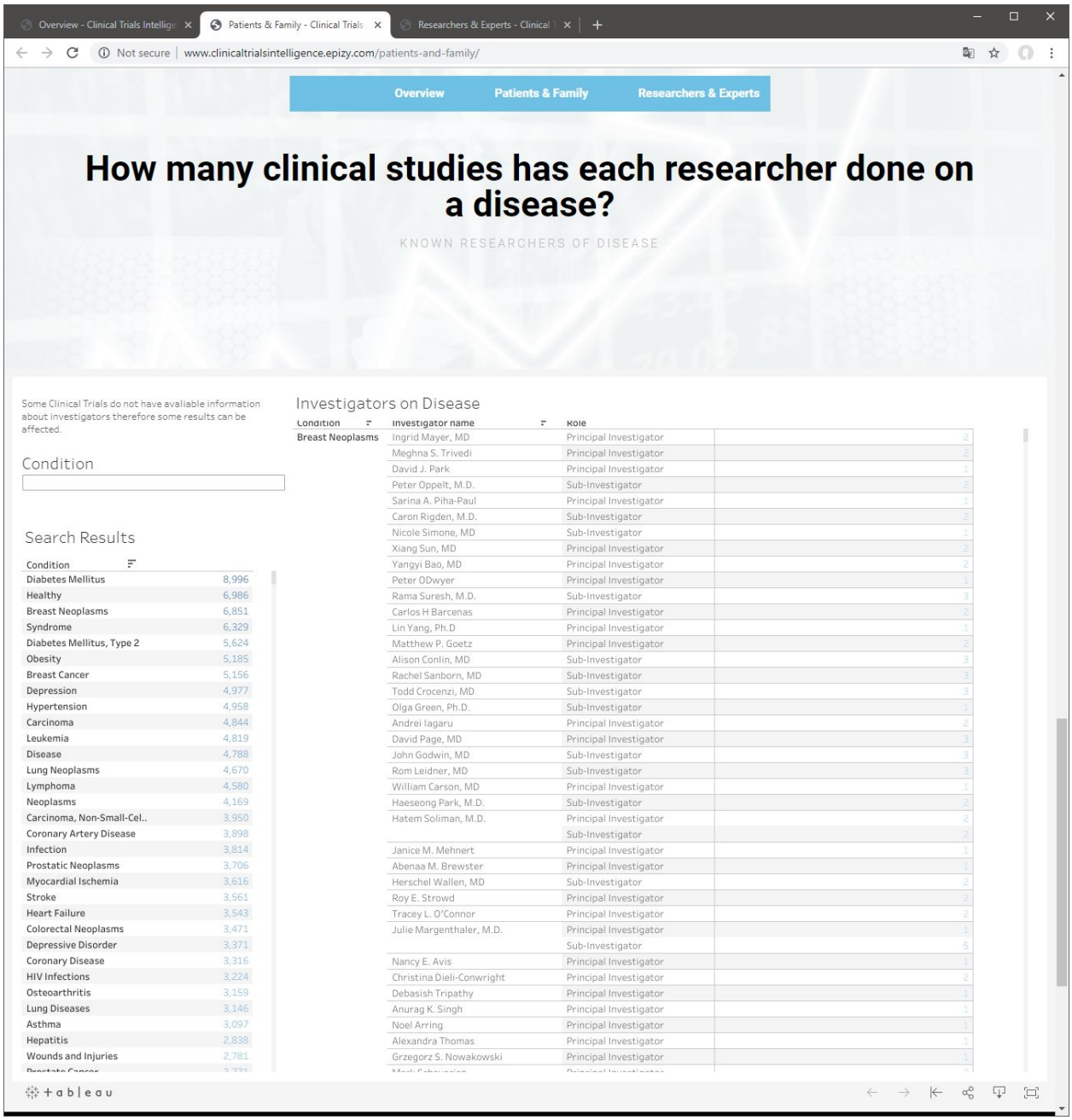


Patients & Family



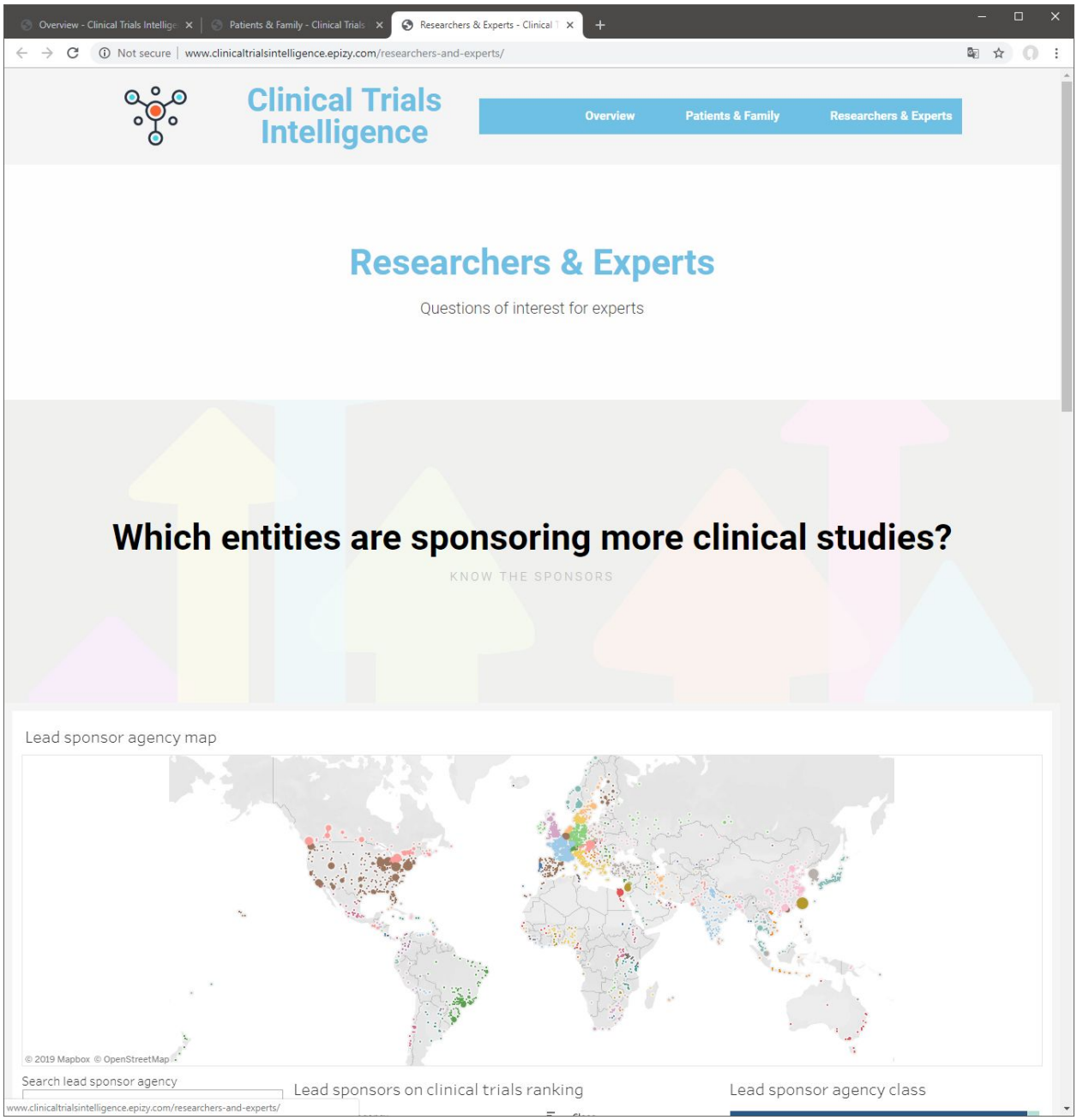
Patients & Family - Q4_conditionEvo

Patients & Family - Q5_conditionWhere

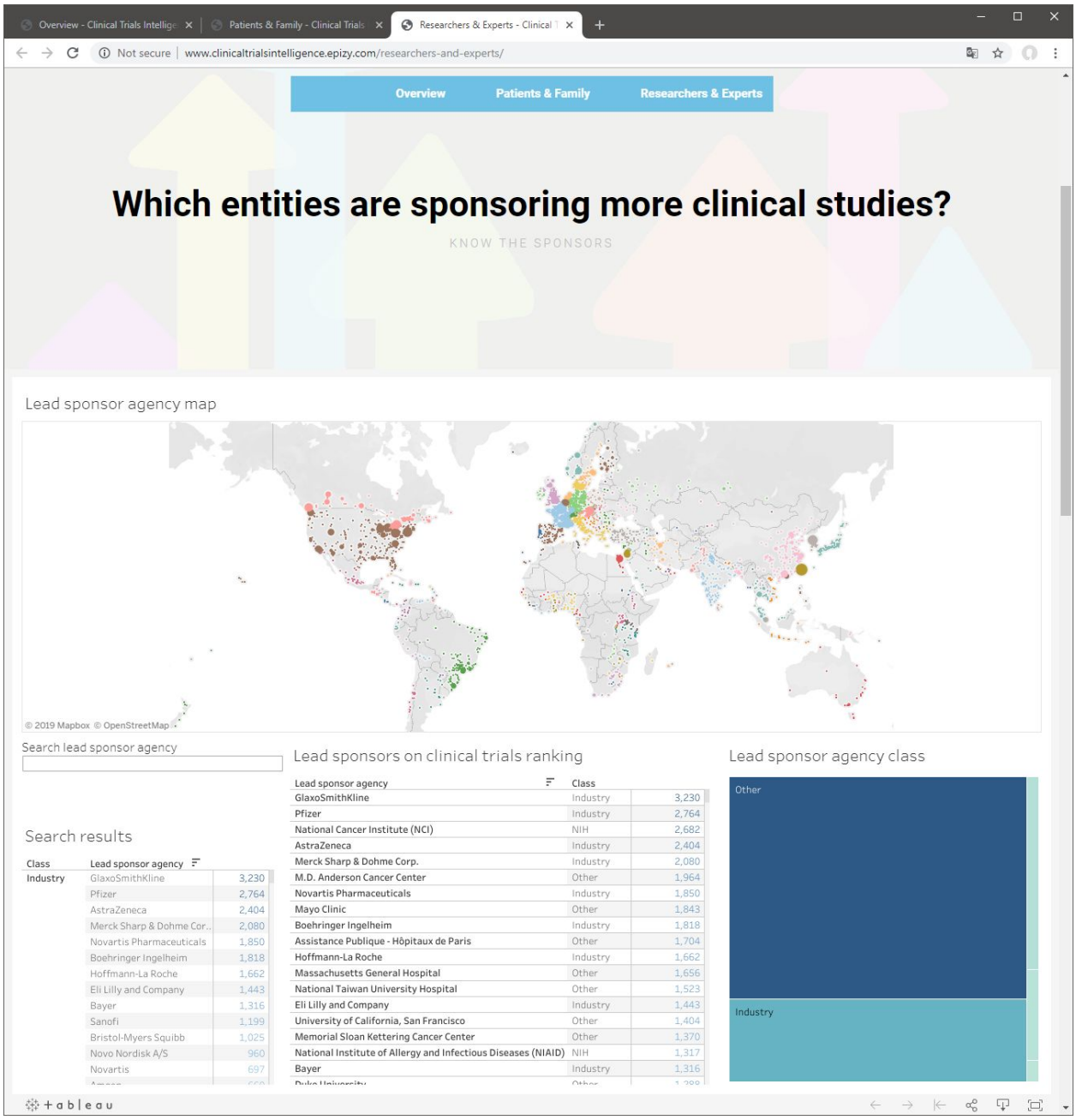


Patients & Family - Q6_conditionInvestigator

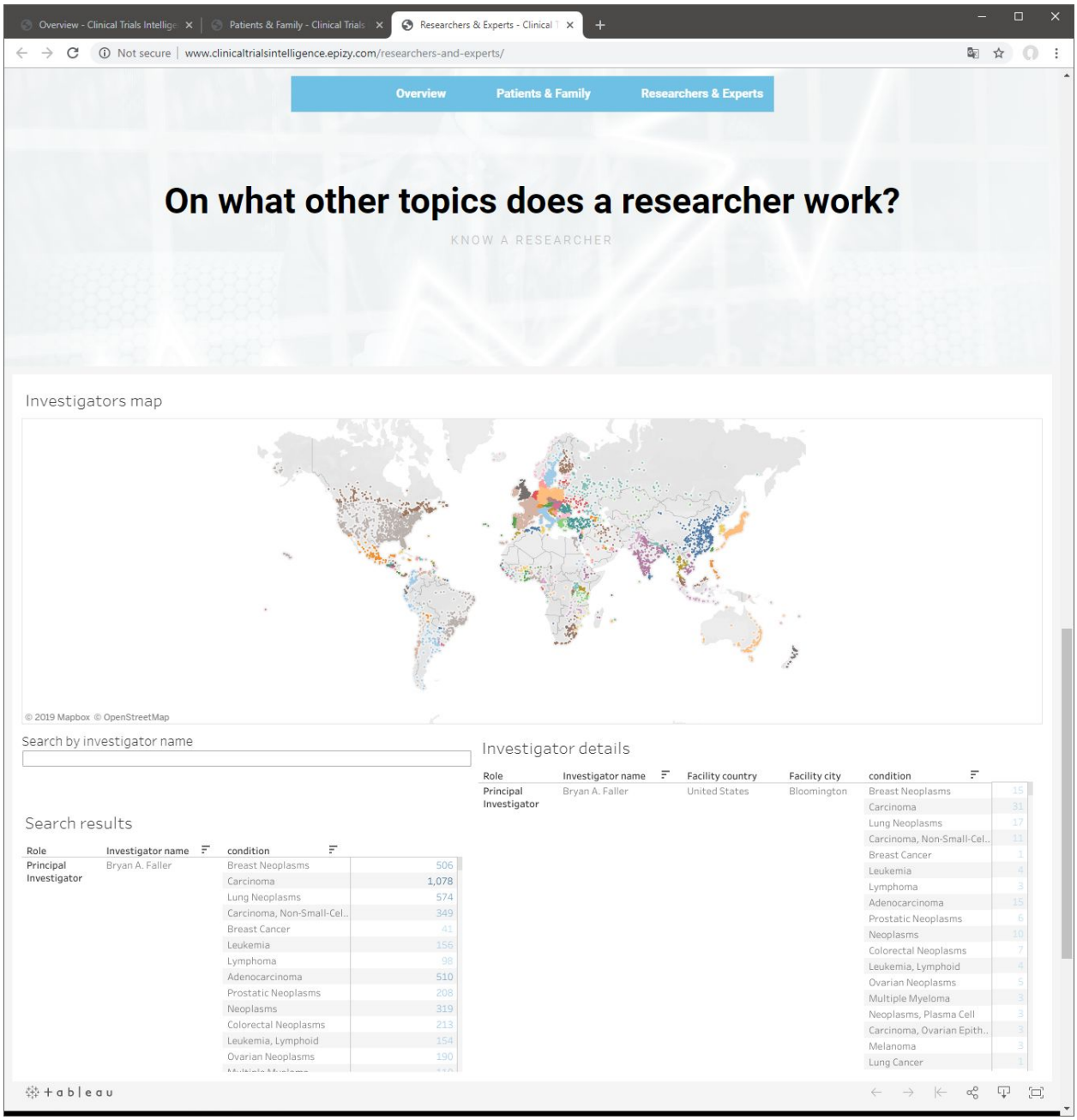
Researchers & Experts



Researchers & Experts

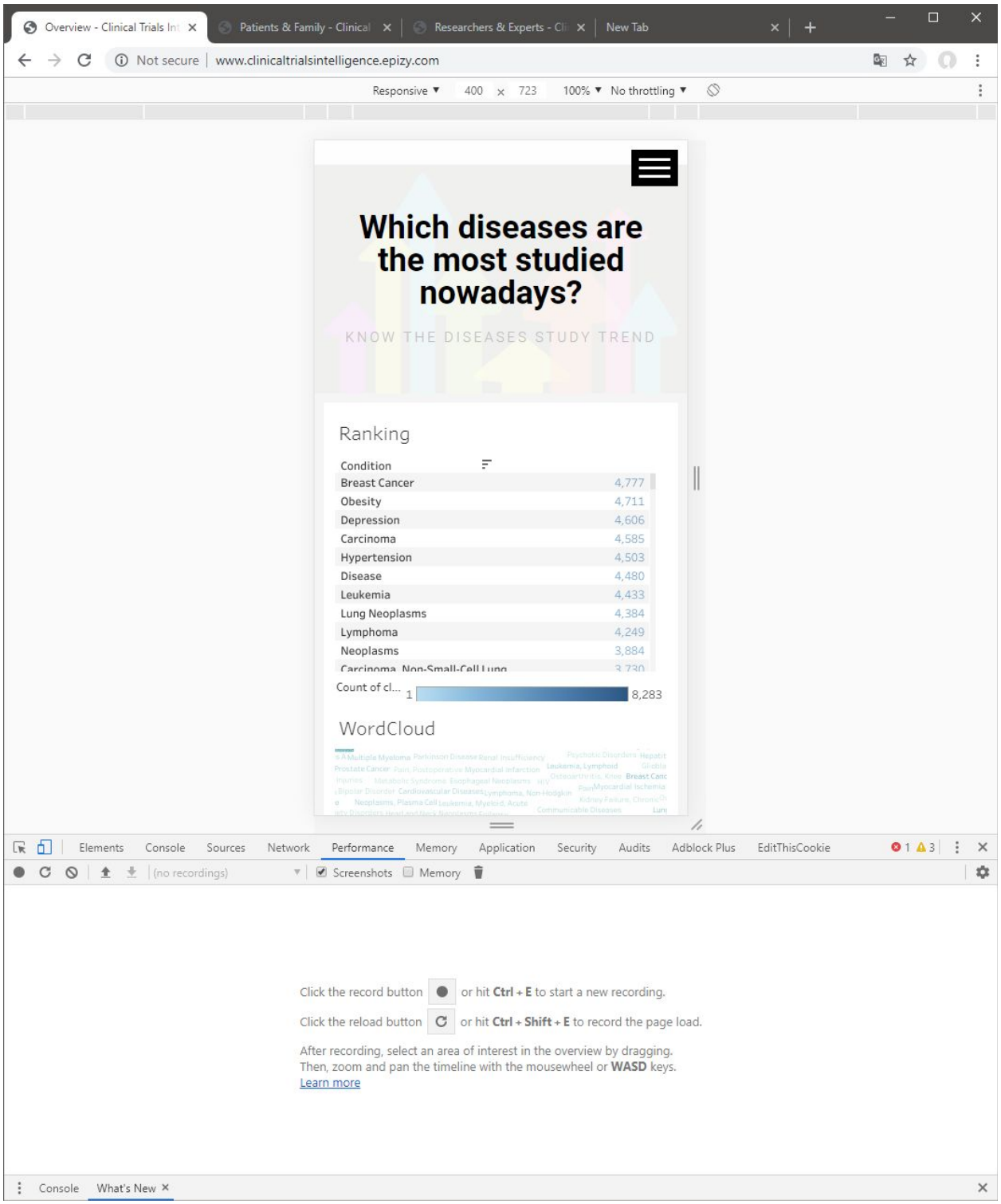


Researchers & Experts - Q7_leadSponsorAgencies

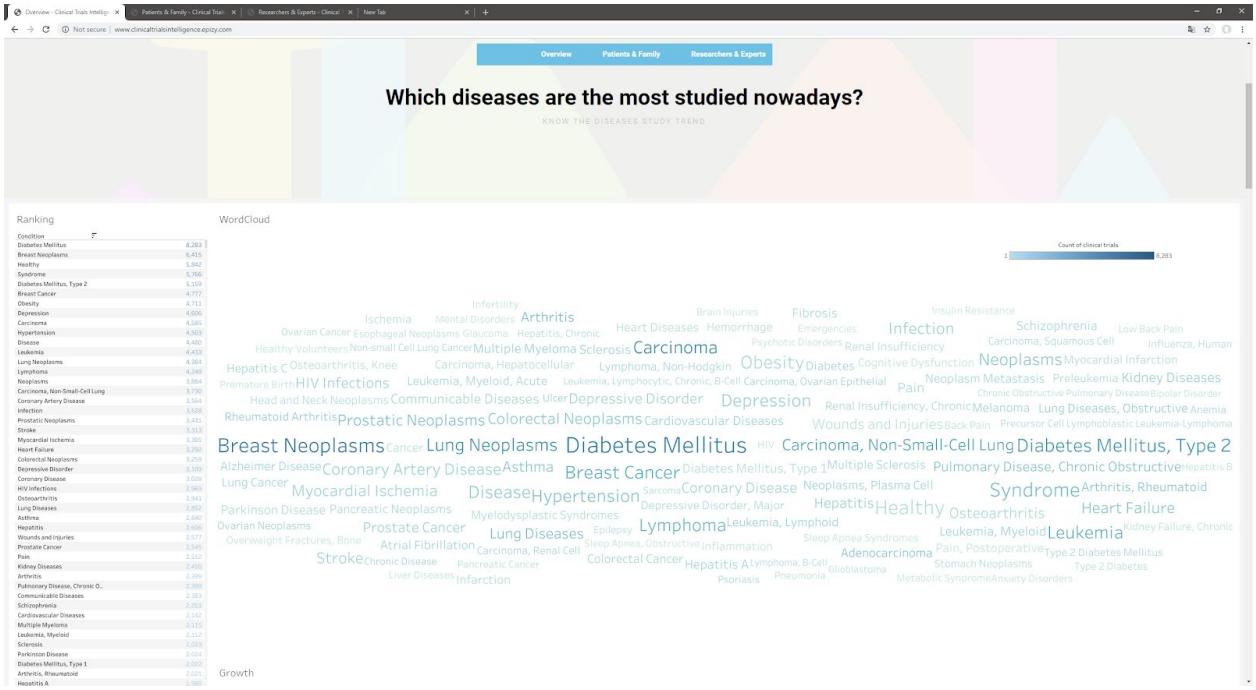


Researchers & Experts - Q9_investigatorsWork

Apéndice 2.1 - Ejemplos de adaptación de las vistas



Overview - q1_studiedDiseasesNow adaptada a smartphone.



Overview - q1_studiedDiseasesNow adaptada a pantalla completa de ordenador.