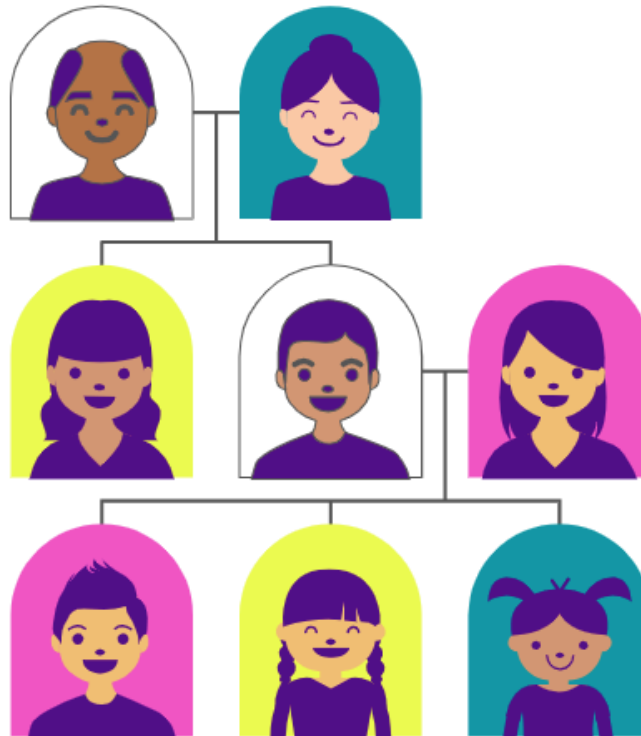


Proyecto:



Árbol Genealógico

Natalia Bottaioli, Rosana García, Leonela Pereira y Rodrigo Souza.

|                                                                                                      |           |
|------------------------------------------------------------------------------------------------------|-----------|
| <b>1 Introducción.....</b>                                                                           | <b>2</b>  |
| 1.1 Objetivo.....                                                                                    | 3         |
| 1.2 Acerca de este documento.....                                                                    | 3         |
| <b>2 Análisis de sistemas existentes.....</b>                                                        | <b>4</b>  |
| 2.1 Herramientas y tipos de datos manejados.....                                                     | 4         |
| 2.1.1 Formato para intercambio de datos genealógicos.....                                            | 6         |
| 2.2 Herramientas para el modelado y almacenamiento de relaciones genealógicas en bases de datos..... | 7         |
| 2.2.1 Soluciones de bases de datos utilizadas.....                                                   | 8         |
| 2.2.2 Bases de datos SQL.....                                                                        | 9         |
| 2.2.3 Base de datos NoSQL.....                                                                       | 10        |
| 2.2.3.1 Bases de datos de grafos.....                                                                | 10        |
| 2.2.3.2 Bases de datos orientadas a columnas.....                                                    | 11        |
| 2.2.4 Ejemplo de representación de un árbol genealógico.....                                         | 12        |
| 2.2.4.1 Representación en una BDR.....                                                               | 13        |
| 2.2.4.2 Representación en una base de datos gráfica (Neo4J).....                                     | 17        |
| 2.2.4.3 Representación en Cassandra.....                                                             | 18        |
| 2.2.5 Ejemplo de recuperación de datos en un árbol genealógico.....                                  | 19        |
| 2.3 Búsquedas avanzadas y recuperación de información en sistemas genealógicos existentes.....       | 20        |
| 2.3.1 Filtros Personalizados.....                                                                    | 21        |
| 2.3.1.1 Filtros Basados en Atributos.....                                                            | 21        |
| 2.3.1.2 Filtros Basados en Relaciones.....                                                           | 21        |
| 2.3.2 Consultas Personalizadas.....                                                                  | 21        |
| 2.3.2.1 Consultas SQL.....                                                                           | 21        |
| 2.3.2.2 Interfaz de Consulta Gráfica.....                                                            | 22        |
| <b>3 Caso de Estudio.....</b>                                                                        | <b>23</b> |
| <b>4 Comparación: Gramps vs FamilySearch.....</b>                                                    | <b>25</b> |
| <b>5 Conclusiones.....</b>                                                                           | <b>26</b> |
| <b>6 Trabajo Futuro.....</b>                                                                         | <b>27</b> |
| <b>7 Referencias.....</b>                                                                            | <b>28</b> |
| <b>8 Referencias Figuras.....</b>                                                                    | <b>29</b> |
| <b>9 Referencias Tablas.....</b>                                                                     | <b>30</b> |
| <b>10 Anexo: Visualizaciones.....</b>                                                                | <b>31</b> |
| 10.1 Gramps.....                                                                                     | 31        |
| 10.2 FamilySearch.....                                                                               | 33        |
| <b>11 Anexo: Arquitectura MyHeritage.....</b>                                                        | <b>35</b> |

# 1 Introducción

La **genealogía** es “el estudio y seguimiento de la ascendencia y la descendencia de una persona o familia” (*Genealogía*, n.d.). El **árbol genealógico** es la herramienta utilizada para documentar dicho estudio.

Desde que existen los registros civiles de identificación (alrededor de 1870 (*Genealogía*, n.d.)), aunque hay países, como Francia, donde los registros son anteriores), los documentos en los que se basa la genealogía son las actas de nacimiento, matrimonio, defunción y, de existir, de reconocimiento. Anterior a la existencia de dichas oficinas públicas, el registro de los hechos vitales de las personas se llevaba a cabo en las iglesias y parroquias y, en ese caso, se registraban bautismos, matrimonios y sepulturas. Fue obligatorio para las iglesias creadas desde el siglo XVI en adelante el llevar estos libros. Sin embargo, existen registros eclesiásticos desde el siglo XII en adelante.

Con el desarrollo de las tecnologías que permiten digitalizar documentos, sumado al interés de organizaciones internacionales como la UNESCO en preservar archivos históricos (*Convention Concerning the Protection of the World Cultural and Natural Heritage*, 1972), la necesidad de almacenar grandes volúmenes de información asociada al estado civil de las personas ha crecido. Asimismo, la aparición de empresas y organizaciones sin fines de lucro que proveen herramientas para construir árboles genealógicos ha dotado tanto a genealogistas como a personas sin formación especializada de aplicativos donde aportar información que permite la creación de bases de datos cada vez más grandes.

Un ejemplo de tales organizaciones es la estadounidense FamilySearch (*Family Search*, n.d.), fundada por la Iglesia de Jesucristo de los Santos de los Últimos Días, que actualmente cuenta con registros de más de 1500 millones de personas, de las cuales casi 900 mil pertenecen a una misma componente conexas<sup>1</sup> del árbol genealógico de la humanidad que la mencionada organización tiene por objetivo construir. Para lograr un objetivo así de ambicioso, el almacenamiento de los registros y relaciones entre registros representa un desafío grande. Asimismo, el contar con cantidades tan enormes de registros atrae a más usuarios a los sistemas con lo cual el conjunto de consultas (lecturas) y escrituras recurrentes pasa a ser del orden de los miles por día.

Motivados por este contexto, este trabajo se plantea los objetivos descritos en la sección que sigue.

---

<sup>1</sup> Fuente: entrevista que integrantes del equipo de proyecto mantuvieron con un ingeniero de FamilySearch.

## 1.1 Objetivo

Estudiar los Sistemas de Gestión de Datos Genealógicos con Capacidades de Búsqueda Avanzada implica, en particular, estudiar cómo:

1. **almacenar** datos biográficos de personas asociados a documentos y construir relaciones genealógicas entre individuos;
2. **facilitar la recuperación eficiente** de información genealógica mediante consultas avanzadas.

Es por esto que el principal objetivo de este proyecto puede describirse como el estudio de cómo resolver los dos puntos mencionados anteriormente. Para esto, se definen una serie de **preguntas guía de la investigación**:

1. ¿Cuáles son los **sistemas existentes** de gestión de datos genealógicos y **qué características** tienen?
2. ¿Qué **técnicas y herramientas** se utilizan para modelar y almacenar **relaciones genealógicas** en bases de datos?
3. ¿Cómo se aborda la **búsqueda avanzada** y la **recuperación de información** en sistemas genealógicos existentes?

## 1.2 Acerca de este documento

A lo largo del presente documento, se intenta responder a las preguntas guía planteadas. En la [Sección 2](#) se aborda el análisis de las soluciones existentes, recorriendo las herramientas genealógicas, las herramientas para el modelado y almacenamiento de información genealógica en bases de datos, y la búsqueda avanzada y recuperación de información en sistemas genealógicos existentes. En la [Sección 3](#) se describe una aplicación de recuperación de información en un árbol genealógico a través del uso de una herramienta en concreto y en la [Sección 4](#) se incluye una tabla comparativa entre Gramps y FamilySearch. Finalmente, en la [Sección 5](#) se resumen las conclusiones y la [Sección 6](#) plantea el trabajo futuro y posibles mejoras a los sistemas existentes.

## 2 Análisis de sistemas existentes

Esta sección pretende responder a la pregunta guía número 1: ¿**Cuáles** son los **sistemas existentes** de gestión de datos genealógicos y **qué características** tienen?

El estudio de la historia familiar es cada vez más accesible gracias a la gran diversidad de plataformas en línea dedicadas a la gestión de datos genealógicos. Entre las empresas y organizaciones que se dedican a la gestión de estos datos se destacan **Ancestry**, **MyHeritage**, **Geni**, **FamilySearch**, **Findmypast**, **WikiTree** y **GeneaFinder**, todas ellas con plataformas en línea. Estas herramientas permiten a los usuarios explorar su herencia, construir árboles genealógicos detallados, y descubrir información valiosa a través de registros históricos y análisis de ADN.

Además de las plataformas en línea, otra herramienta que se encontró fue **Gramps**, un programa de código abierto para la genealogía que se puede descargar y utilizar de forma gratuita.

El tipo de datos que se puede llegar a cargar y/o acceder depende de la plataforma. A continuación se presenta una breve descripción de los datos que maneja cada uno de estos sitios.

### 2.1 Herramientas y tipos de datos manejados

En esta subsección se listan y describen las principales herramientas y el tipo de datos que almacenan y/o a las que tienen acceso desde sus plataformas.

**Ancestry** (Ancestry, n.d.): Empresa genealógica estadounidense con sede en Lehi, Utah. Entre los datos que maneja se encuentran los siguientes: registros de nacimiento, matrimonio y defunción; censos y listas de votantes; registros de migración y viajes; registros militares; directorios de ciudades; historias de iglesias; registros de impuestos y terrenos; referencias y diccionarios; árboles genealógicos públicos; colecciones especiales de afroamericanos, indígenas, judíos, registros del estado de Nueva York y cuáqueros. Además, proporciona información genética basada en pruebas de ADN, incluyendo descendencia étnica y regiones geográficas.

**MyHeritage** (MyHeritage, n.d.): Inicialmente fundada en Israel, MyHeritage ofrece acceso a 17 mil millones de expedientes genealógicos, incluyendo actas de nacimiento, matrimonio, defunción, censos, registros militares, de inmigración, anuarios, y copias originales de documentos. Las categorías de búsqueda abarcan censos, listas de votantes, árboles genealógicos, fotos, periódicos, libros, datos públicos, directorios, guías, historias, memorias, biografías, registros de gobierno y mapas. Destacan su colección histórica de periódicos con 120 millones de páginas y todos los censos del Reino Unido de 1841 a 1901, así como cientos

de millones de registros de nacimientos, matrimonios y defunciones del Reino Unido. La plataforma añade un millón de nuevos expedientes históricos y perfiles a los árboles genealógicos diariamente. Al igual que Ancestry, MyHeritage proporciona información genética basada en pruebas de ADN, incluyendo ascendencia étnica y regiones geográficas.

**FamilySearch** (FamilySearch, n.d.): Empresa propiedad de la Iglesia de Jesucristo de los Santos de los Últimos Días, con sede en Utah. Los datos provienen de diversas categorías de registros históricos, que incluyen censos y listas de población, directorios, genealogías recopiladas, historia, migración y naturalización, registros misceláneos, registros de nacimiento, matrimonio y defunción, datos de origen étnico, registros variados, periódicos, registros de servicio militar y registros de sucesiones testamentarias y tribunales. El sitio es de uso gratuito, pero a cambio de poder utilizar el mismo, FamilySearch declara que toda la información que le sea proporcionada, pasa a ser de su propiedad (*Licencia Y Derechos Otorgados*, 2023).

**WikiTree** (WikiTree, n.d.): Contiene información sobre registros históricos, árboles genealógicos, datos de ADN. Es un sitio de uso gratuito. Al aceptar los términos y condiciones de uso del sitio, suceden dos cosas (*Terms of Service*, 2009). Por un lado, al publicar nuestros datos en wikitree, otorgamos una licencia a Interesting.com (sitio que aloja a WikiTree) para usar, copiar, distribuir, mostrar, realizar y transmitir el contenido. Por otro lado, al ser una wiki, otros usuarios tienen permiso de edición sobre nuestros datos. Sobre este segundo aspecto, podemos tener un poco más de control, ya que cada usuario puede configurar lo que se llama una Lista de Confianza. En esa lista se agrega a aquellos usuarios que permitimos tengan acceso a nuestra información. Si un usuario no forma parte de esa Lista de Confianza, la información que puede ver de los otros usuarios dependerá del nivel de privacidad que cada usuario configure en su perfil (*Trusted List*, n.d).

**Geneafinder** (Geneafinder, n.d.): Sitio francés que contiene registros históricos, árboles genealógicos, documentos familiares.

**Geni** (Geni, n.d.): Sitio web comercial con sede en California, propiedad de MyHeritage desde 2012. Almacena perfiles familiares, árboles genealógicos, fotos familiares, y otros documentos.

**Findmypast** (Findmypast, n.d.): Sitio web británico propiedad de DC Thomson. Contiene registros históricos de censos, del ejército, periódicos, de la iglesia católica, registros de matrimonio, nacimiento y defunción. También cuenta con datos de migración. Los datos son de Gales e Inglaterra.

**Gramps** (Gramps, n.d.): Aplicación que ofrece una amplia gama de características para la gestión de datos genealógicos que incluyen la capacidad de crear perfiles individuales para cada miembro de la familia, registrar eventos importantes como nacimientos, matrimonios y defunciones, y establecer relaciones familiares a través de árboles genealógicos interactivos. También permite a los usuarios editar y personalizar los datos genealógicos, añadiendo notas, imágenes y otros detalles relevantes. Está disponible para varias plataformas, incluyendo

Windows, macOS, Linux y también cuenta con una versión web. Gramps es un proyecto comunitario, creado y desarrollado por genealogistas.

### 2.1.1 Formato para intercambio de datos genealógicos

Existe un formato estándar muy utilizado para la transferencia de datos genealógicos, denominado Genealogical Data Communication (GEDCOM), que permite la importación y exportación de datos genealógicos facilitando la colaboración y el intercambio de información entre usuarios de las diferentes plataformas. (Repositorio De árboles Genealógicos En Una Base De Datos NoSql, 2015).

Ejemplo de dato GEDCOM:

```
0 @I0054@ INDI
1 NAME Walter Matthew /Marín/
2 TYPE birth
2 GIVN Walter Matthew
2 SURN Marín
1 SEX M
1 BIRT
2 TYPE Birth of Marín, Walter Matthew
2 DATE 29 DEC 1893
2 PLAC Worthington, MN, USA
2 ADDR
3 CITY Worthington
3 STAE MN
3 CTRY USA
1 DEAT
2 TYPE Death of Marín, Walter Matthew
2 DATE 16 JAN 1969
2 PLAC Worthington, MN, USA
2 ADDR
3 CITY Worthington
3 STAE MN
3 CTRY USA
1 BURI
2 TYPE Burial of Marín, Walter Matthew
2 DATE 18 JAN 1969
2 PLAC Worthington, MN, USA
2 ADDR
3 CITY Worthington
3 STAE MN
3 CTRY USA
1 FAMC @F0035@
2 PEDI birth
1 FAMS @F0034@
1 SOUR @S0003@
1 CHAN
2 DATE 26 JUL 2007
3 TIME 05:34:25
```

@I0054@: Este es el identificador único del individuo, que permite referirse a él de manera única en el archivo GEDCOM.

NAME: Contiene el nombre del individuo, dividido en partes (GIVN para el nombre propio y SURN para el apellido).

SEX: Indica el sexo del individuo (en este caso, masculino).

BIRT, DEAT y BURI: Son eventos relacionados con el nacimiento, la muerte y el entierro del individuo, respectivamente. Cada evento incluye detalles como el tipo de evento, la fecha, el lugar y la dirección.

FAMC y FAMS: Estas etiquetas identifican las familias a las que pertenece el individuo, ya sea como hijo (FAMC) o como cónyuge (FAMS). Los identificadores proporcionados (@F0035@ y @F0034@) hacen referencia a las familias correspondientes.

PEDI: Describe la relación de parentesco del individuo con sus padres biológicos (birth indica que es el hijo biológico).

SOUR: Hace referencia a la fuente de información sobre este individuo, identificada por un identificador único (@S0003@).

CHAN: Indica cambios o actualizaciones en el registro del individuo, incluyendo la fecha y hora de la modificación.

En conclusión, las plataformas antes mencionadas manejan diversos tipos de datos, incluyendo registros de nacimiento, matrimonio y defunción, censos, listas de votantes, registros militares, de inmigración y viajes, directorios de ciudades, anuarios, periódicos, y datos públicos. También, proporcionan acceso a colecciones especiales y datos específicos de regiones geográficas y grupos étnicos. Todos estos datos pueden ser intercambiados por estas plataformas a través del formato GEDCOM.

Las plataformas Ancestry, MyHeritage, Geni, Findmypast y GeneaFinder, ofrecen funciones limitadas de forma gratuita, pero tienen opciones de pago para acceder al conjunto completo de servicios y características. Gramps, FamilySearch y WikiTree son completamente gratis. Sin embargo, mientras que en Gramps los datos son propiedad del usuario, en FamilySearch y WikiTree los datos pasan a ser propiedad de ellos.

## 2.2 Herramientas para el modelado y almacenamiento de relaciones genealógicas en bases de datos

Esta sección pretende responder a la pregunta guía número 2: ¿Qué técnicas y herramientas se utilizan para **modelar y almacenar relaciones genealógicas** en bases de datos?

El **modelado de datos** es un proceso que consiste en **identificar entidades y sus relaciones**. (*¿Qué Es El Modelado De Datos?*, n.d.) En esta sección se describen los paradigmas de almacenamiento de datos que se utilizan para modelar y almacenar relaciones genealógicas en bases de datos.



### 2.2.1 Soluciones de bases de datos utilizadas

Se presentan brevemente las bases de datos utilizadas por algunos de los sitios previamente mencionados, a saber: **Ancestry**, **MyHeritage**, **FamilySearch** y **Gramps**.

**Ancestry** (How Ancestry.com Branched Out And Embraced Scale, n.d.): Utiliza una combinación de tecnologías para gestionar y procesar grandes cantidades de datos relacionados con los árboles genealógicos en su plataforma.

Opera dos centros de datos, uno en Utah y otro en Denver, Colorado, con miles de servidores y más de 10 PT de datos en producción, con redundancia en almacenamiento auxiliar para seguridad. Utiliza una combinación de base de datos relacional y Hadoop<sup>2</sup> para almacenar datos de árboles genealógicos y genéticos. La búsqueda se realiza a través de un motor de búsqueda propio llamado Gravity, optimizado específicamente para buscar y acceder a miles de millones de documentos escaneados e indexados. Los documentos escaneados son almacenados en arreglos de disco Isilon, y los datos generados por los usuarios se almacenan en arreglos BlueArc. La empresa también utiliza MySQL para sus bases de datos relacionales, con técnicas de particionamiento para mejorar el rendimiento y la escalabilidad. La base de datos de árboles genealógicos se ejecuta en servidores ProLiant DL980 de HP.

**MyHeritage** (MyHeritage Cassandra Meetup 2016, n.d.): Utiliza la base de datos MySQL Sharding<sup>3</sup>, base de datos particionada, horizontalmente distribuida en más de 650 servidores (denominados *shards*), que no se conocen entre sí. Tiene un buen tiempo de respuesta cuando se consulta por una única familia (acceso a un solo *shard*) pero se complejiza cuando se necesita acceder a múltiples *shards* a la vez, por ejemplo para mostrar resultados de búsqueda o coincidencias de perfiles provenientes de muchos árboles genealógicos. Para resolver ese problema, a MySQL se suma el uso de Cassandra. En el anexo 11 se adjunta una imagen de la arquitectura de datos.

**FamilySearch** (FamilySearch DataStax, n.d.): utilizan DataStax Enterprise por su escalabilidad y rendimiento. DataStax es una base de datos NoSQL y vectorial construida sobre Apache Cassandra.

**Gramps**: Utilizó, en versiones anteriores, BSDDb (Berkeley Software Distribution database) una base de datos no relacional y jerárquica, pero desde la versión 5.1 utiliza SQLite.

---

<sup>2</sup> Hadoop (*¿Qué Es Hadoop?*, n.d.) es una estructura de software de código abierto que proporciona almacenamiento masivo de cualquier tipo de datos con enorme poder de procesamiento. Mucho más económico por ser de código abierto.

<sup>3</sup> *Sharding* (Particionando Bases De Datos Horizontalmente (Sharding), 2022) es una técnica de particionamiento horizontal para bases de datos relacionales que divide y distribuye los datos en diferentes servidores de bases de datos (cada uno denominado *shard*) separados físicamente, cada uno con el mismo tipo de *hardware*, motor y estructura de datos para generar un nivel de rendimiento similar. La característica principal de estos servidores es que **no se conocen entre sí**, lo que los diferencia de otros enfoques de escalado horizontal como la replicación o clústeres de BD.

En conclusión, el almacenamiento y la recuperación de los datos es un sistema complejo en el que se utiliza mucha tecnología y diferentes tipos de bases de datos, lo cual tiene sentido debido a la cantidad y diversidad de datos que se manejan.

### 2.2.2 Bases de datos SQL

Una **base de datos relacional** (BDR) (*Modelo Relacional Conceptos Básicos Introducción a Las Bases De Datos*, n.d.) es un sistema de almacenamiento de datos que organiza la información en tablas. Estas tablas están relacionadas entre sí mediante claves. El Structured Query Language (SQL) es el lenguaje estándar para interactuar con bases de datos relacionales. Permite realizar consultas, insertar, actualizar y eliminar datos, así como gestionar la estructura de la base de datos.

Para entender el funcionamiento de una BDR, necesitamos entender los siguientes conceptos:

**Tabla:** Una tabla es una colección de datos organizados en filas y columnas. Cada fila representa un **registro** único, y cada columna representa un **atributo** del registro.

**Fila:** También conocida como “tupla”, representa un solo elemento o registro en la tabla.

**Columna:** Representa un atributo o campo de los registros almacenados en la tabla.

**Clave primaria:** Es un campo o una combinación de campos que identifica de manera única a cada registro en una tabla.

**Clave foránea:** Es un campo en una tabla que refiere a la clave primaria de otra tabla, estableciendo una relación entre ambas.

**Relación:** Las BDR utilizan claves primarias y foráneas para establecer relaciones entre las tablas. Una relación puede ser de uno a uno (1:1), uno a muchos (1:N) o muchos a muchos (N:N).

El uso de BDR para el registro de árboles genealógicos tiene varias ventajas. Para empezar, las BDR están diseñadas para almacenar datos en tablas, lo que proporciona una estructura bien organizada para almacenar información como nombres, fechas, nombres de padres, etc. Esta estructura facilita la consulta y actualización de la información. También permiten establecer relaciones entre diferentes conjuntos de datos, y esto puede ayudar a mantener la integridad de los datos. Además, la estructura de la base permite que la consulta sea más eficiente, todo el código reside del lado de la BD y esto permite sacar información compleja rápidamente.

Sin embargo, podemos tener algunos problemas referentes a la escalabilidad. A medida que la cantidad de datos crece, las bases de datos relacionales pueden volverse menos eficientes. Esto se debe a que la estructura tabular de las bases de datos puede volverse difícil de manejar con grandes volúmenes de datos.

### 2.2.3 Base de datos NoSQL

En esta sección se analizan dos abordajes NoSQL: las bases de datos de grafos (Neo4j) y las bases de datos orientadas a columnas (Cassandra).

#### 2.2.3.1 Bases de datos de grafos

Las **bases de datos de grafos (BDG)** (*Bases De Datos De Grafos*, n.d.) son un modelo de bases de datos que utiliza la teoría de grafos para almacenar registros, utilizando **vértices** y **aristas** para representar las entidades y sus relaciones, respectivamente. Se vienen usando en diversas áreas como el análisis de redes sociales, sistemas de recomendación, y en la detección de fraudes. Son especialmente útiles cuando la información puede ser fácilmente transformada en nodos y relaciones, así como cuando los datos ingresan en varios formatos, ya que **son más flexibles en la representación de datos** que las bases de datos relacionales.

Para modelar **árboles genealógicos**, las bases de datos de grafos son particularmente efectivas, ya que **los datos se pueden representar naturalmente** como **nodos y relaciones**. Las bases de datos relacionales y algunos tipos de bases de datos NoSQL tienen dificultades para modelar relaciones de manera eficiente, y pueden aumentar la complejidad al realizar consultas de distintas fuentes. Además, **consultas recíprocas** pueden incurrir en un alto costo computacional en estos sistemas.

Un ejemplo de consulta recíproca en el contexto de relaciones familiares podría ser determinar si dos personas son primos. En una base de datos de grafos, donde las relaciones familiares se modelan como nodos y relaciones, podríamos tener un nodo para cada persona y relaciones que representen los diferentes tipos de relaciones familiares (padre, madre, hijo, hija, hermano, hermana, etc.).

Supongamos que queremos determinar si Juan y María son primos. En una consulta recíproca, primero buscaríamos todos los nodos que son hermanos de los padres de Juan y luego comprobaríamos si María es hija de alguno de esos nodos. Luego, repetiríamos el proceso pero considerando a los padres de María, intentando ver si Juan es hijo de alguno de esos nodos. Si en ambas búsquedas encontramos una conexión entonces podemos asegurar que los datos son consistentes y que Juan y María son primos.

Esta consulta puede ser compleja de expresar en una base de datos relacional debido a la necesidad de realizar múltiples joins y comparaciones, lo que puede resultar en un costo computacional elevado y una consulta más complicada. En una base de datos de grafos, la consulta puede ser más intuitiva y eficiente, ya que simplemente seguimos las relaciones directamente desde los nodos de Juan y María para determinar su parentesco.

Neo4j (*Neo4j*, n.d.) es una de las bases de datos de grafos más populares en la actualidad. Implementada en Java, utiliza el lenguaje de consultas Cypher, que es declarativo y similar a

SQL. Esto permite realizar consultas especificando los resultados deseados en lugar de tener que especificar un algoritmo para obtener un resultado.

Neo4j también ofrece diferentes APIs para lenguajes de programación populares como Java y Python. Viene en dos ediciones: Community y Enterprise. La edición Enterprise, que es paga, ofrece mayores capacidades, como un tamaño máximo de grafos más grande y capacidades de paralelización.

Comparativamente, las bases de datos de grafos tienen ventajas sobre las bases de datos relacionales y algunas bases de datos NoSQL en la modelación de relaciones complejas, como las requeridas para modelar árboles genealógicos. Sin embargo, es importante considerar las limitaciones de las bases de datos de grafos.

En entornos con un alto volumen de escritura o análisis de grandes conjuntos de datos, pueden no ser la mejor opción. Las operaciones de escritura masiva pueden ser más lentas y menos eficientes en comparación con las bases de datos relacionales o ciertas bases de datos NoSQL diseñadas para la escalabilidad horizontal. Además, las bases de datos de grafos pueden requerir más memoria y almacenamiento para gestionar las conexiones entre los nodos, lo que puede convertirse en un cuello de botella en sistemas de gran escala.

Además, la elección entre bases de datos de grafos, relacionales y NoSQL depende de los requisitos específicos del proyecto, como el tipo de datos, el rendimiento esperado y la complejidad de las consultas.

#### 2.2.3.2 Bases de datos orientadas a columnas

Como se vio en la sección dedicada a bases de datos relacionales, en este tipo de bases de datos la información se almacena en tablas y se utilizan claves foráneas para hacer referencia a datos que se almacenan en otras tablas. Las consultas se hacen, básicamente, a través de joins de tablas.

Apache Cassandra también almacena su información en tablas. Su principal diferencia con las bases de datos relacionales es que su modelo de datos está diseñado alrededor de las consultas que se desea hacer, y optimizado para el mismo fin. Es decir, al diseñar una base de datos a ser implementada en Apache Cassandra, se tiene en cuenta las consultas que se querrá realizar y se diseña la base en base a ellas. El lenguaje utilizado para realizar estas consultas es el CQL (Cassandra Query Language). Una diferencia grande por la cual Cassandra no soporta el modelado habitual utilizado en bases de datos relacionales es que el lenguaje CQL no soporta joins.

En Cassandra el modelado de datos está guiado por las queries que se desea realizar. De hecho, las consultas acceden a una única tabla, lo cual implica que todas las entidades involucradas en una consulta deben estar en la misma tabla para hacer el acceso a datos

(lecturas) muy rápido. Esto implica, en particular, que un mismo dato puede estar presente en más de una tabla. Este proceso se conoce como denormalización. La duplicación de información y un alto rendimiento en los procesos de escritura son usados para alcanzar una alta performance de lectura.

A nivel de arquitectura, Cassandra está concebida como una base de datos distribuida que almacena información en un clúster de nodos. Se utilizan claves de partición para particionar datos en los diferentes nodos.

En conclusión, Cassandra está orientado a consultas, es decir, sus tablas se diseñan pensando en cuáles serán las consultas más habituales de un cierto sistema. Esto es así porque está pensado para ser extremadamente eficiente en tiempos de lectura y, por lo tanto, no se realizan consultas complejas sino que, para una cierta consulta, lo que se termina haciendo es recuperar una línea de una cierta tabla. Esto implica que un mismo dato aparezca en más de una tabla, y se tenga lo que se conoce como consistencia eventual, que quiere decir que, si bien los datos alcanzan un punto de consistencia, esto no sucede inmediatamente, especialmente al modificar un cierto dato que pueda estar presente en más de una tabla.

En el caso de aplicaciones muy masivas, como la de la estadounidense FamilySearch, el tener tiempos de lectura rápidos es fundamental, especialmente dado el volumen de datos y usuarios que tiene el sistema (de acuerdo a información provista por la mencionada empresa, tienen 4 billones de **lecturas al día** vs 129 millones de **escrituras**).

#### 2.2.4 Ejemplo de representación de un árbol genealógico

Luego de haber explorado diferentes formas que existen en la actualidad para almacenar datos de relaciones genealógicas, nos propusimos hacer un breve ejercicio que nos sirviera para ejemplificar cómo se representa un mismo árbol genealógico en las distintas BD. Dada una familia hipotética, Figura 2, compuesta por padre, madre y dos hijos, a continuación se muestra como sería una posible representación para el caso de usar una BDR y para el caso de usar una BD basada en grafos.

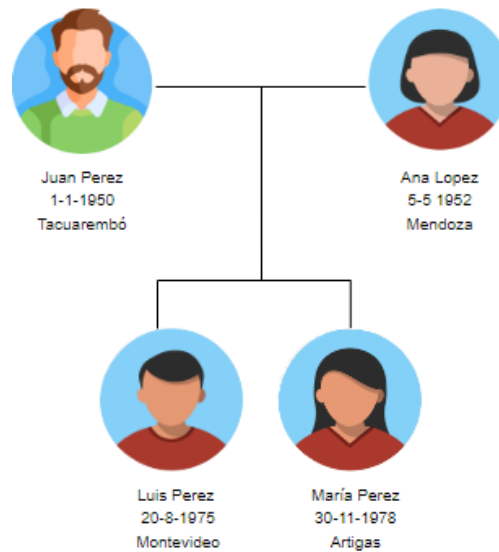


Figura 1. Familia hipotética

#### 2.2.4.1 Representación en una BDR

En este caso elegimos representar la familia hipotética con el modelo de BD de la Tabla 1 y tener dos tablas principales: Persona y Relación. La tabla Persona contendría la información básica de cada individuo, y la tabla Relación definiría las conexiones entre estas personas (por ejemplo, quién es padre/madre de quién). Las relaciones son de un árbol ascendente y además se pueden obtener datos de cónyuges y hermanos.

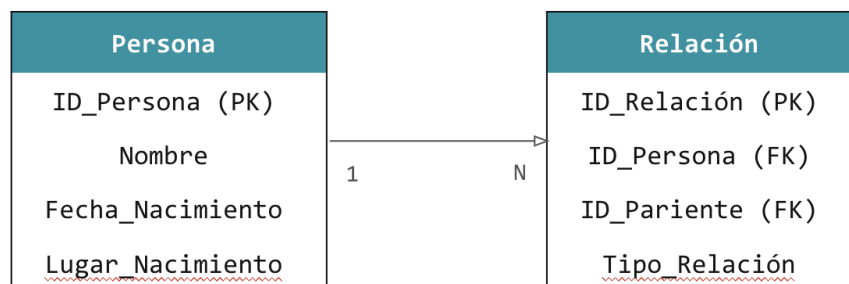


Tabla 1. Posible modelo BD.

El código requerido para crear estas tablas en SQL es el siguiente:

Creación de la Tabla Persona:

```
CREATE TABLE Persona (  
    ID_Persona INT PRIMARY KEY,  
    Nombre VARCHAR(100),  
    Fecha_Nacimiento DATE,  
    Lugar_Nacimiento VARCHAR(100)  
);
```

Inserción de datos en la Tabla Persona:

```
INSERT INTO Persona (ID_Persona, Nombre, Fecha_Nacimiento,  
Lugar_Nacimiento)  
VALUES  
(1, 'Juan Pérez', '1950-01-01', 'Tacuarembó'),  
(2, 'Ana López', '1952-05-05', 'Mendoza'),  
(3, 'Luis Pérez', '1975-08-20', 'Montevideo'),  
(4, 'María Pérez', '1978-11-30', 'Artigas');
```

Creación de la Tabla Relación:

```
CREATE TABLE Relación (  
    ID_Relación INT PRIMARY KEY,  
    ID_Persona INT,  
    ID_Pariente INT,  
    Tipo_Relación VARCHAR(50),  
    FOREIGN KEY (ID_Persona) REFERENCES Persona(ID_Persona),  
    FOREIGN KEY (ID_Pariente) REFERENCES Persona(ID_Persona)  
);
```

Inserción de datos en la Tabla Relación:

```
INSERT INTO Relación (ID_Relación, ID_Persona, ID_Pariente, Tipo_Relación)
VALUES
(1, 1, 3, 'Padre'),
(2, 2, 3, 'Madre'),
(3, 1, 4, 'Padre'),
(4, 2, 4, 'Madre'),
(5, 1, 2, 'Cónyuge'),
(6, 3, 4, 'Hermana');
```

Tabla: Persona

| ID_Persona (PK) | Nombre      | Fecha_Nacimiento | Lugar_Nacimiento |
|-----------------|-------------|------------------|------------------|
| 1               | Juan Pérez  | 1950-01-01       | Tacuarembó       |
| 2               | Ana López   | 1952-05-05       | Mendoza          |
| 3               | Luis Pérez  | 1975-08-20       | Montevideo       |
| 4               | María Pérez | 1978-11-30       | Artigas          |

Tabla 2. Tabla persona.

Tabla: Relación

| ID_Relacion (PK) | ID_Persona (FK) | ID_Pariente (FK) | Tipo_Relacion |
|------------------|-----------------|------------------|---------------|
| 1                | 1               | 3                | Padre         |
| 2                | 2               | 3                | Madre         |
| 3                | 1               | 4                | Padre         |
| 4                | 2               | 4                | Madre         |
| 5                | 1               | 2                | Cónyuge       |
| 6                | 3               | 4                | Hermana       |

Tabla 3. Tabla relación.



- Tabla 2. **Persona:** Esta tabla contiene los datos básicos de cada persona en el árbol genealógico.
  - a. **ID\_Persona:** Clave primaria que identifica de manera única a cada persona.
  - b. **Nombre:** Nombre de la persona.
  - c. **Fecha\_Nacimiento:** Fecha de nacimiento de la persona.
  - d. **Lugar\_Nacimiento:** Lugar de nacimiento de la persona.
- Tabla 3. **Relación** Esta tabla define las relaciones entre las personas.
  - a. **ID\_Relacion:** Clave primaria que identifica de manera única a cada relación.
  - b. **ID\_Persona:** Clave foránea que refiere a la persona en cuestión.
  - c. **ID\_Pariente:** Clave foránea que refiere al pariente de la persona.
  - d. **Tipo Relación:** Tipo de relación (por ejemplo, Padre, Madre, Hijo, Hija, Hermano, Hermana, etc).

Datos que salen de la tablas:

- **Persona:**
  - a. Juan Pérez (ID 1) nació el 1 de enero de 1950 en Tacuarembó.
  - b. Ana López (ID 2) nació el 5 de mayo de 1952 en Mendoza.
  - c. Luis Pérez (ID 3) nació el 20 de agosto de 1975 en Montevideo.
  - d. María Pérez (ID 4) nació el 30 de noviembre de 1978 en Artigas.
- **Relación:**
  - a. Juan Pérez (ID 1) es el padre de Luis Pérez (ID 3).
  - b. Ana López (ID 2) es la madre de Luis Pérez (ID 3).
  - c. Juan Pérez (ID 1) es el padre de María Pérez (ID 4).
  - d. Ana López (ID 2) es la madre de María Pérez (ID 4).
  - e. Juan Pérez (ID 1) es el cónyuge de Ana López (ID 2) .
  - f. Luis Pérez (ID 3) es el hermano de María Pérez (ID 4).

De esta forma podríamos almacenar y consultar relaciones familiares de manera estructurada. También es posible incluir otros detalles referentes tanto a personas como a relaciones, según sea necesario.

#### 2.2.4.2 Representación en una base de datos gráfica (Neo4J)

En este caso para el mismo árbol y sus tipos de relaciones tenemos 4 nodos, uno por persona, cada uno con sus propiedades como nombre, fecha de nacimiento y lugar de nacimiento, y las relaciones entre los nodos definen los vínculos familiares, como "Padre" y "Madre".

Creación de Nodos:

```
CREATE (p1:Persona {id: 1, nombre: 'Juan Pérez', fecha_nacimiento:
'1950-01-01', lugar_nacimiento: 'Tacuarembó'})
CREATE (p2:Persona {id: 2, nombre: 'Ana López', fecha_nacimiento:
'1952-05-05', lugar_nacimiento: 'Mendoza'})
CREATE (p3:Persona {id: 3, nombre: 'Luis Pérez', fecha_nacimiento:
'1975-08-20', lugar_nacimiento: 'Montevideo'})
CREATE (p4:Persona {id: 4, nombre: 'María Pérez', fecha_nacimiento:
'1978-11-30', lugar_nacimiento: 'Artigas'})
```

Creación de Relaciones:

```
MATCH (p1:Persona {id: 1}), (p3:Persona {id: 3})
CREATE (p1)-[:RELACION {tipo: 'Padre'}]->(p3)

MATCH (p2:Persona {id: 2}), (p3:Persona {id: 3})
CREATE (p2)-[:RELACION {tipo: 'Madre'}]->(p3)

MATCH (p1:Persona {id: 1}), (p4:Persona {id: 4})
CREATE (p1)-[:RELACION {tipo: 'Padre'}]->(p4)

MATCH (p2:Persona {id: 2}), (p4:Persona {id: 4})
CREATE (p2)-[:RELACION {tipo: 'Madre'}]->(p4)

MATCH (p1:Persona {id: 1}), (p2:Persona {id: 2})
CREATE (p1)-[:RELACION {tipo: 'Cónyuge'}]->(p2)

MATCH (p3:Persona {id: 3}), (p4:Persona {id: 4})
CREATE (p3)-[:RELACION {tipo: 'Hermano'}]->(p4)
```

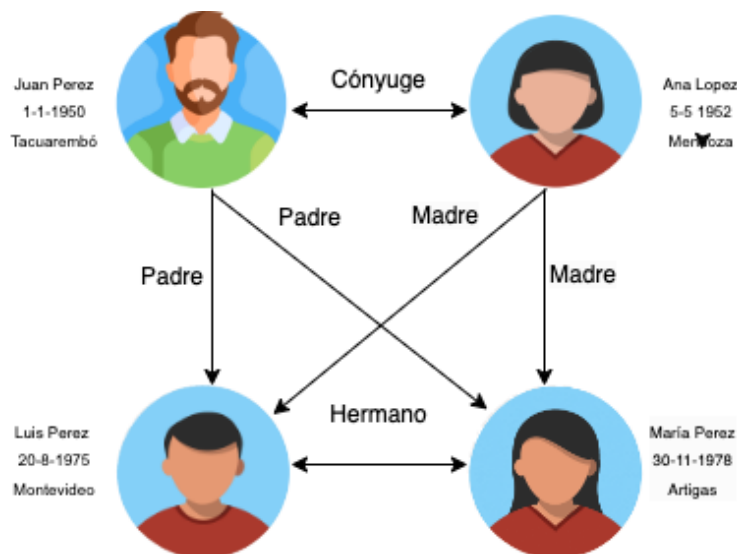


Figura 2. Posible árbol genealógico en Neo4j.

#### 2.2.4.3 Representación en Cassandra

Se incluye aquí el mismo modelado, ahora utilizando Cassandra.

Creación de la Tabla Persona:

```
CREATE TABLE persona (
    id_persona UUID PRIMARY KEY,
    nombre TEXT,
    fecha_nacimiento DATE,
    lugar_nacimiento TEXT
);
```

Creación de la Tabla Relación:

```
CREATE TABLE relación (
    id_persona UUID,
    id_pariente UUID,
    tipo_relacion TEXT,
    PRIMARY KEY (id_persona, id_pariente)
);
```

El código para agregar datos a la tabla Persona:

```
INSERT INTO persona (id_persona, nombre, fecha_nacimiento,
lugar_nacimiento)
VALUES
(UUID_de_Luis_Perez, 'Juan Pérez', '1950-01-01', 'Tacuarembó'),
(UUID_de_Ana_Lopez, 'Ana López', '1952-05-05', 'Mendoza'),
(UUID_de_Luis_Perez, 'Luis Pérez', '1975-08-20', 'Montevideo'),
(UUID_de_Maria_Perez, 'María Pérez', '1978-11-30', 'Artigas');
```

El código para agregar datos a la tabla Relación:

```
INSERT INTO relación (id_persona, id_pariente, tipo_relacion)
VALUES
(UUID_de_Luis_Perez, UUID_de_Juan_Perez, 'Padre'),
(UUID_de_Luis_Perez, UUID_de_Ana_Lopez, 'Madre'),
(UUID_de_Maria_Perez, UUID_de_Juan_Perez, 'Padre'),
(UUID_de_Maria_Perez, UUID_de_Ana_Lopez, 'Madre'),
(UUID_de_Ana_Lopez, UUID_de_Juan_Perez, 'Cónyuge'),
(UUID_de_Maria_Perez, UUID_de_Luis_Perez, 'Hermano');
```

### 2.2.5 Ejemplo de recuperación de datos en un árbol genealógico

Si, para el mismo ejemplo de la sección anterior, quisiéramos contar la cantidad de hermanos de María Pérez en cada una de las opciones de representación del árbol, las consultas serían así:

**En SQL:**

1. Primero se obtiene el ID de Persona de María Pérez con SELECT en la tabla **Persona**:

```
id_maria = (SELECT ID_Persona FROM Persona WHERE Nombre = 'María
Pérez');
```

2. Luego, se utiliza ese ID para contar cuántas relaciones de tipo 'Hermano' tiene María Pérez en la tabla **Relación**:

```
SELECT COUNT(*) AS Numero_Hermanos
FROM Relacion
WHERE ID_Persona = id_maria AND Tipo_Relacion = 'Hermano';
```

### En Cypher:

1. Se encuentra el nodo de María Pérez usando su nombre:  
`MATCH (p:Persona {Nombre: 'María Pérez'}) RETURN p;`
2. Se cuenta cuántos nodos están relacionados con ella mediante la relación HERMANO\_DE: .  
`MATCH (p:Persona {Nombre: 'María Pérez'}) - [:HERMANO_DE] → (hermano)  
RETURN COUNT(hermano) AS Numero_Hermanos;`

### En CQL:

1. Primero se obtiene el UUID de Persona de María Pérez con SELECT en la tabla **Persona**:

```
SELECT id_persona  
FROM persona  
WHERE nombre = 'María Pérez'
```

2. Luego, se utiliza ese ID para contar cuántas relaciones de tipo 'Hermano' tiene María Pérez en la tabla **Relación**:

```
SELECT COUNT(*) AS Numero_de_Hermanos  
FROM relacion  
WHERE id_persona = 'UUID_de_Maria_Perez' AND tipo_relacion = 'Hermano';
```

Observar que, en este caso, CQL y SQL prácticamente no difieren en la consulta, dado que no se trata de consultas que impliquen un JOIN en SQL. Para consultas más complejas, sí diferirían estas consultas dado que CQL no soporta, justamente, los JOINS.

## 2.3 Búsquedas avanzadas y recuperación de información en sistemas genealógicos existentes

Esta sección pretende responder a la siguiente pregunta guía planteada al inicio del trabajo: ¿Cómo se aborda la búsqueda avanzada y la recuperación de información en sistemas genealógicos existentes?

Los filtros y consultas personalizadas son esenciales en los sistemas de datos genealógicos para facilitar la recuperación de información específica y relevante. Estas herramientas

permiten a los usuarios definir criterios precisos para buscar y filtrar datos en grandes bases de datos genealógicas. A continuación, se detallan los aspectos clave de los filtros y consultas personalizadas en estos sistemas.

### 2.3.1 Filtros Personalizados

Para determinar los filtros aplicables, se revisaron dos herramientas genealógicas de uso gratuito: FamilySearch y Gramps. Esta revisión permitió identificar que se pueden aplicar filtros basados en atributos y en relaciones.

#### 2.3.1.1 Filtros Basados en Atributos

Los **filtros basados en atributos** (*Cómo Utilizar Los Filtros De Búsqueda En FamilySearch.org*, n.d.) permiten a los usuarios restringir los resultados de búsqueda según características específicas de los registros genealógicos, tales como:

- **Nombres y Apellidos:** Buscar personas con nombres o apellidos específicos.
- **Fechas de Nacimiento, Matrimonio y Muerte:** Filtrar registros en un rango de fechas.
- **Lugares:** Filtrar por ubicaciones geográficas específicas donde ocurrieron eventos.
- **Eventos:** Filtrar por tipo de evento (nacimiento, matrimonio, defunción, etc.).

#### 2.3.1.2 Filtros Basados en Relaciones

Los sistemas genealógicos pueden filtrar registros basados en las relaciones entre individuos (*Gramps - Filter*, n.d.):

- **Relaciones Familiares:** Filtrar por relaciones específicas como padres, hijos, cónyuges, etc.
- **Árbol Genealógico:** Restringir resultados a miembros de un subárbol genealógico específico.

### 2.3.2 Consultas Personalizadas

#### 2.3.2.1 Consultas SQL

Para usuarios con conocimientos técnicos, algunos sistemas genealógicos, como Gramps, permiten el uso de consultas SQL (*Gramps - Addon:QueryGramplet*, n.d.) directamente en la base de datos para realizar búsquedas complejas, como por ejemplo:

- **Consultas Selectivas:** Obtener registros que cumplan con múltiples condiciones.
- **Joins:** Combinar datos de múltiples tablas para obtener información detallada sobre individuos y sus relaciones.
- **Agregación:** Calcular estadísticas o agrupar datos por atributos específicos.

### 2.3.2.2 Interfaz de Consulta Gráfica

Algunas herramientas genealógicas, como Ancestry (Ancestry - *Search*, n.d.), ofrecen una interfaz gráfica para construir consultas personalizadas sin necesidad de conocimientos de SQL:

- **Construcción Visual de Consultas:** Los usuarios pueden arrastrar y soltar campos y operadores para construir consultas complejas.
- **Asistentes de Búsqueda:** Guías paso a paso para definir criterios de búsqueda y filtros.

### 3 Caso de Estudio

Una vez realizada la revisión de las tecnologías utilizadas para árboles genealógicos, a sugerencia de la docente responsable del curso se decide explorar la posibilidad de analizar alguna solución con Neo4j. Intentamos levantar varios repositorios de GitHub<sup>4</sup> para realizar pruebas, pero nos enfrentamos a diversas dificultades y no logramos hacerlo con éxito. Optamos, entonces, por instalar Gramps, solución abierta que permite verificar, y tener un mejor control sobre las funcionalidades que ofrece.

Con Gramps instalado procedimos a analizar algunas búsquedas de datos. Hay dos formas nativas de encontrar datos en Gramps: Búsqueda y Filtro, que funcionan de manera muy diferente (*Gramps 5.2 Wiki Manual*, n.d.).

**Búsqueda:** La búsqueda examina la base de datos tal como aparece en las filas y columnas en la pantalla. Por ejemplo, si se configuró la visualización del nombre para mostrar "Apellido, Nombre", entonces se pueden buscar coincidencias en todas las filas del tipo "Smith, J". Si se cambia la forma en que se muestran los nombres, entonces cambia la forma en se hace la coincidencia (por ejemplo, "John Smith").

**Filtro:** No se limita a lo que se ve en la pantalla, sino que examina los datos reales, en lugar de solo lo que se muestra en la vista. Por ejemplo, el filtro de nombre intentará hacer coincidir en cualquier campo de nombre único (nombre, apellido, prefijo, etc.) de todos los nombres (primario y alternativo), pero solo un campo por nombre; no se puede hacer coincidir el campo del nombre y el campo del apellido en el mismo nombre. Es decir, si se filtra por "John", se obtienen coincidencias de personas con el nombre "John" pero también aquellos con el apellido "Johnson". No se pueden filtrar combinaciones de nombres y apellidos.

Aunque, a priori, el filtro pareciera no ser mejor que la búsqueda, en realidad hay que tener presente las siguientes consideraciones:

- El Filtro busca también en los nombres alternativos, mientras que la Búsqueda se limita a buscar en el nombre principal (el que se muestra). Por eso, si se hace un Filtro por "Smith", podría verse personas listadas que no parecen coincidir. Pero si se accede a ver los detalles de esa persona, seguramente tenga un nombre alternativo que contiene "smith" (ejemplo: "Goldsmith").
- El Filtro permite la utilización de "expresiones regulares". Por lo tanto, se pueden encontrar todos los nombres que comienzan con "B" y terminan en "ship": "B.\*ship". No se puede hacer eso con la Búsqueda.

---

<sup>4</sup> Los repositorios de Neo4j que intentamos probar fueron los siguientes:

- <https://github.com/frankienamdi/family-tree>
- <https://github.com/sylhare/family-tree>
- <https://github.com/larien/family-tree>



- La Búsqueda solo hará coincidir lo que es visible. Si un nombre o texto es demasiado grande para verse en la lista que aparece debajo de la barra superior, entonces no se encontrará.
- Todos los Filtros utilizan coincidencia sin distinguir mayúsculas y minúsculas; "Ship" coincidirá con "ship", "SHIP" o "ShIp".

**Consultas a la BDR:** Se puede hacer consultas SQL directamente a la base de datos utilizando una herramienta externa a Gramps, ya que Gramps no cuenta con una consulta a la base desde su interfaz usuario.

## 4 Comparación: Gramps vs FamilySearch

La Tabla 4 presenta un resumen comparativo entre las funcionalidades verificadas en Gramps y aquellas que, según lo declarado o verificable desde la perspectiva del usuario, se encuentran en FamilySearch.

|                                                      | Gramps                                                                                                                                       | Family Search                                                         |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Web                                                  | ✓                                                                                                                                            | ✓                                                                     |
| Local                                                | ✓                                                                                                                                            | ✗                                                                     |
| Permite exportar datos                               | ✓                                                                                                                                            | ✗                                                                     |
| Permite importar datos                               | ✓                                                                                                                                            | ✓                                                                     |
| Formato de datos soportados                          | Gramps XML, Gramps Package (Portable Gramps XML), <b>GEDCOM</b> , GeneWeb's GW format, Web Family Tree (. WFT) format, vCard, vCalendar, CSV | GEDCOM X                                                              |
| Base de datos                                        | Base de datos relacional                                                                                                                     | Base de datos no relacional                                           |
| Formatos de visualizaciones disponibles <sup>5</sup> | Árbol, abanico, mapamundi                                                                                                                    | Árbol y abanico                                                       |
| Propietario de los datos                             | Usuario                                                                                                                                      | Plataforma                                                            |
| Código abierto                                       | ✓                                                                                                                                            | ✗                                                                     |
| Controles de datos                                   | Fechas incorrectas, relaciones familiares inválidas, formato de archivos                                                                     | No hay controles para agregar datos, y para borrar un dato es difícil |
| Eliminar datos                                       | ✓                                                                                                                                            | ✓                                                                     |
| Derecho al olvido                                    | ✓                                                                                                                                            | ✗                                                                     |

Tabla 4. Comparación entre dos herramientas. Gramps vs Family Search.

<sup>5</sup> Ver Sección 10 Anexo: visualizaciones.

## 5 Conclusiones

Actualmente, existen varias soluciones para la gestión de datos genealógicos, casi todas basadas en la web, excepto Gramps, que ofrece una aplicación de escritorio. Algunas, como FamilySearch, wikitree y Gramps, son gratuitas, mientras que el resto requieren una suscripción paga. Entre las gratuitas, FamilySearch y wikitree se convierten en propietarias de los datos proporcionados por los usuarios, mientras que con Gramps, al estar instalado en tu propio PC o servidor, mantienes la propiedad de tus datos. Estas soluciones utilizan diversas fuentes de información, incluyendo partidas de nacimiento, matrimonio y defunción, censos, registros militares, migratorios y de periódicos.

La elección de la arquitectura para modelar árboles genealógicos debe considerar tanto la representación de las relaciones entre individuos como la administración de grandes volúmenes de datos. Las empresas investigadas emplean arquitecturas complejas que combinan diversas tecnologías. Algunas utilizan bases de datos relacionales para modelar las relaciones, combinadas con bases de datos distribuidas como Cassandra y tecnologías como Hadoop o MySQL Sharding, para resolver aspectos de rendimiento, seguridad, escalabilidad y redundancia. Esto es esencial, dado que empresas como MyHeritage y FamilySearch almacenan millones de documentos en sus bases de datos.

En el modelado de relaciones, analizamos tres enfoques: bases de datos relacionales, Neo4j y Cassandra. Las bases de datos relacionales destacan por su robustez y capacidad para realizar consultas complejas, aunque enfrentan desafíos de escalabilidad. Las bases de datos de grafos como Neo4j, que inicialmente parecían la opción más natural para representar árboles genealógicos, no son las más utilizadas en la práctica debido a sus limitaciones en operaciones masivas de escritura. Por otro lado, Cassandra es extremadamente eficiente en tiempos de lectura, siendo ideal para aplicaciones con gran volumen de datos y usuarios.

Finalmente, se analizó la búsqueda avanzada y la recuperación de información en algunas de las herramientas. Los filtros pueden basarse en atributos, relaciones familiares y subárboles. Herramientas como Gramps permiten consultas SQL personalizadas, proporcionando un acceso más directo y flexible a los datos almacenados.

La realidad genealógica es difícil de representar de manera única y efectiva. No existe una solución única ni una que sea superior a las demás; la elección de la tecnología depende de múltiples factores específicos a cada caso. Esto se refleja en la diversidad de tecnologías y bases de datos utilizadas para resolver este problema.

## 6 Trabajo Futuro

En el futuro, se propone implementar una serie de mejoras y nuevas funcionalidades para optimizar los sistemas genealógicos. Estas incluyen:

- **Uso de Neo4J como base de datos:** Adoptar Neo4J para el almacenamiento de árboles genealógicos permitirá modelar la realidad de manera más eficiente. Si bien almacenar un árbol genealógico muy grande puede presentar desafíos en términos de escalabilidad, Neo4J puede ser utilizado en algunas partes o subárboles para aprovechar las ventajas de la teoría de grafos. Esto facilitará la representación de relaciones complejas y permitirá utilizar una amplia gama de algoritmos basados en grafos, mejorando el rendimiento y la capacidad de análisis de los sistemas genealógicos.
- **Exportación de datos en FamilySearch:** Se recomienda que FamilySearch permita la exportación de datos. Actualmente, FamilySearch se beneficia de los datos proporcionados por los usuarios, pero no ofrece una forma sencilla de obtener los datos recopilados. La implementación de esta funcionalidad promoverá la reciprocidad y mejorará la colaboración entre usuarios y la plataforma.
- **Modernización de la interfaz de Gramps:** Para brindar una mejor experiencia de usuario, es fundamental modernizar la interfaz de Gramps. Una interfaz más intuitiva y atractiva facilitará la navegación y el uso del software, haciendo que la creación y gestión de árboles genealógicos sea más accesible para usuarios de todos los niveles de experiencia.

Estas mejoras no solo potenciarán la funcionalidad y usabilidad de los sistemas genealógicos, sino que también fomentarán una mayor participación y satisfacción de los usuarios.

## 7 Referencias

Ancestry. (n.d.). Ancestry® | Family Tree, Genealogy & Family History Records. Retrieved June 8, 2024, from <https://www.ancestry.com/>

Ancestry - Search. (n.d.). Ancestry. Retrieved June 23, 2024, from <https://www.ancestry.com/search/>

Bases de datos de grafos. (n.d.). [https://eva.fing.edu.uy/pluginfile.php/144359/mod\\_resource/content/7/6\\_graphDBs\\_intro.pdf](https://eva.fing.edu.uy/pluginfile.php/144359/mod_resource/content/7/6_graphDBs_intro.pdf)

Cómo utilizar los filtros de búsqueda en FamilySearch.org. (n.d.). familysearch. <https://www.familysearch.org/es/blog/como-utilizar-los-filtros-de-busqueda-en-familysearch-org>

Convention Concerning the Protection of the World Cultural and Natural Heritage. (1972, november 16). <https://whc.unesco.org/archive/convention-en.pdf>

Example filters. (n.d.). Gramps. Retrieved June 23, 2024, from [https://gramps-project.org/wiki/index.php/Example\\_filters](https://gramps-project.org/wiki/index.php/Example_filters)

familysearch. (n.d.). Retrieved June 8, 2024, from <https://beta.familysearch.org/search/collection/list?count=All>

Family Search. (n.d.). Find your family. Free Genealogy Archive. Retrieved June 25, 2024, from <http://www.familysearch.org>

FamilySearch DataStax. (n.d.). DataStax. Retrieved June 9, 2024, from <https://www.datastax.com/enterprise-success/familysearch>

Filter. (n.d.). Gramps. Retrieved June 23, 2024, from <https://gramps-project.org/wiki/index.php/Filter>

findmypast. (n.d.). Retrieved June 8, 2024, from <https://www.findmypast.co.uk/>

Geneafinder. (n.d.). Geneafinder. Retrieved June 8, 2024, from <https://geneafinder.com/>

Genealogía. (n.d.). Wikipedia. Retrieved June 25, 2024, from <https://es.wikipedia.org/wiki/Genealog%C3%ADa>

Genealogía. (n.d.). Wikipedia. Retrieved June 25, 2024, from <https://es.wikipedia.org/wiki/Genealog%C3%ADa>

Geni. (n.d.). Retrieved June 8, 2024, from <https://www.geni.com/>

Gramps. (n.d.). Gramps. Retrieved June 9, 2024, from [https://gramps-project.org/wiki/index.php/Main\\_page](https://gramps-project.org/wiki/index.php/Main_page)

Gramps 5.2 Wiki Manual. (n.d.). Gramps. Retrieved June 22, 2024, from [https://gramps-project.org/wiki/index.php/Gramps\\_5.2\\_Wiki\\_Manual](https://gramps-project.org/wiki/index.php/Gramps_5.2_Wiki_Manual)

Gramps - Addon:QueryGramplet. (n.d.). Gramps. Retrieved June 23, 2024, from <https://www.gramps-project.org/wiki/index.php/Addon:QueryGramplet>

How Ancestry.com Branched Out And Embraced Scale. (n.d.). <https://www.enterpriseai.news/> Retrieved June 8, 2024, from <https://www.enterpriseai.news/2014/01/09/ancestry-com-branches-embraces-scale/>

Licencia y Derechos otorgados. (2023, September 1). FamilySearch. Retrieved June 27, 2024, from <https://www.familysearch.org/legal/terms>

*Modelo Relacional Conceptos Básicos Introducción a las Bases de Datos.* (n.d.). <https://eva.fing.edu.uy/>  
[https://eva.fing.edu.uy/pluginfile.php/491738/mod\\_resource/content/1/IntroRelacional-art.pdf](https://eva.fing.edu.uy/pluginfile.php/491738/mod_resource/content/1/IntroRelacional-art.pdf)  
*myheritage.* (n.d.). Retrieved June 8, 2024, from [https://www.myheritage.es/help-center?a=Que-expedientes-se-incluyen-en-el-Motor-de-Busqueda-de-Registros-Historicos-de-MyHeritage---id--CzO\\_In6dQKaWOmqa-AgXGQ](https://www.myheritage.es/help-center?a=Que-expedientes-se-incluyen-en-el-Motor-de-Busqueda-de-Registros-Historicos-de-MyHeritage---id--CzO_In6dQKaWOmqa-AgXGQ)  
*MyHeritage Cassandra meetup 2016.* (n.d.). Retrieved June 8, 2024, from <https://es.slideshare.net/slideshow/myheritage-cassandra-meetup-2016/70428683>  
*Neo4j.* (n.d.). Neo4j. Retrieved June 27, 2024, from <https://neo4j.com/docs/getting-started/>  
*Particionando bases de datos horizontalmente (sharding).* (2022, June 14). Amazon AWS. Retrieved June 8, 2024, from <https://aws.amazon.com/es/blogs/aws-spanish/particionando-bases-de-datos-horizontalmente-sharding-con-amazon-rds/>  
*¿Qué es el modelado de datos?* (n.d.). IBM. Retrieved June 25, 2024, from <https://www.ibm.com/es-es/topics/data-modeling>  
*¿Qué es Hadoop?* (n.d.). SAS Institute. Retrieved June 8, 2024, from [https://www.sas.com/es\\_mx/insights/big-data/hadoop.html](https://www.sas.com/es_mx/insights/big-data/hadoop.html)  
*Repositorio de árboles genealógicos en una base de datos NoSql.* (2015, October 26). UPCommons. Retrieved June 15, 2024, from <https://upcommons.upc.edu/bitstream/handle/2117/78834/108622.pdf>  
*Terms of Service.* (2009, June 10). WikiTree. Retrieved June 27, 2024, from <https://www.wikitree.com/about/terms.html>  
*Trusted List.* (n.d.). WikiTree. Retrieved June 27, 2024, from [https://www.wikitree.com/wiki/Help:Trusted\\_List](https://www.wikitree.com/wiki/Help:Trusted_List)  
*wikitree.* (n.d.). youtube. Retrieved June 8, 2024, from <https://www.youtube.com/watch?v=MDZ13G7HSPY&list=TLGGmsjJzWk61EQyNzA1MjAyNA>

## 8 Referencias Figuras

Figuras 1 y 2. Elaboración propia.

Figuras 3, 4, 5 y 6. Elaboración propia a partir de: Gramps. Retrieved June 9, 2024, from [https://gramps-project.org/wiki/index.php/Main\\_page](https://gramps-project.org/wiki/index.php/Main_page)

Figura 7. Elaboración propia a partir de: *Family Search.* (n.d.). Find your family. Free Genealogy Archive. Retrieved June 25, 2024, from <http://www.familysearch.org>

Figura 8 . *myheritage.* (n.d.). Retrieved June 8, 2024, from [https://www.myheritage.es/help-center?a=Que-expedientes-se-incluyen-en-el-Motor-de-Busqueda-de-Registros-Historicos-de-MyHeritage---id--CzO\\_In6dQKaWOmqa-AgXGQ](https://www.myheritage.es/help-center?a=Que-expedientes-se-incluyen-en-el-Motor-de-Busqueda-de-Registros-Historicos-de-MyHeritage---id--CzO_In6dQKaWOmqa-AgXGQ)

## 9 Referencias Tablas.

Tabla 1, 2 , 3 y 4. Elaboración Propia.

# 10 Anexo: Visualizaciones

## 10.1 Gramps

A modo complemento, en este anexo se presentan algunas de las formas de visualización para los árboles genealógicos interactivos que presenta Gramps.

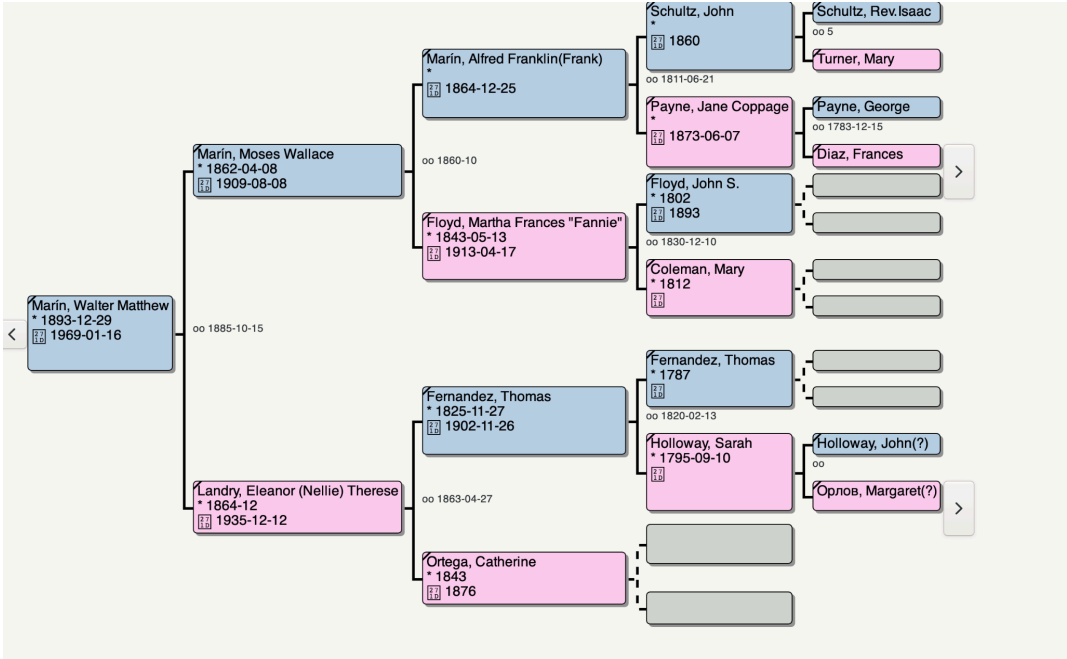


Figura 3. Pedigree Chart



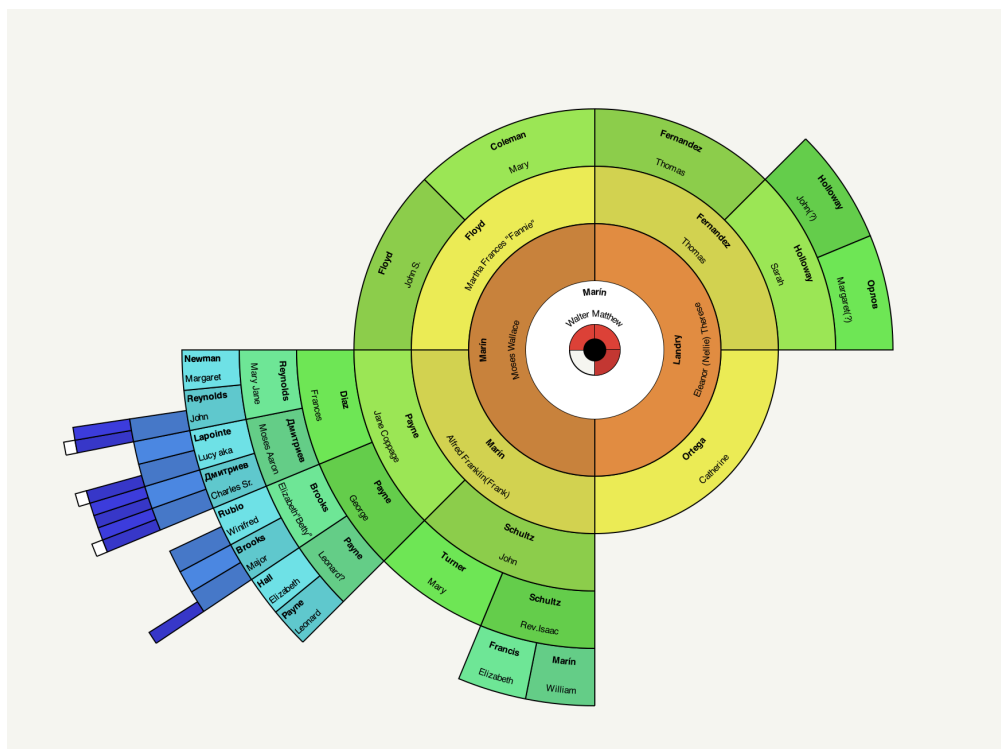


Figura 4. Fan chart

Mapas y geolocalización:

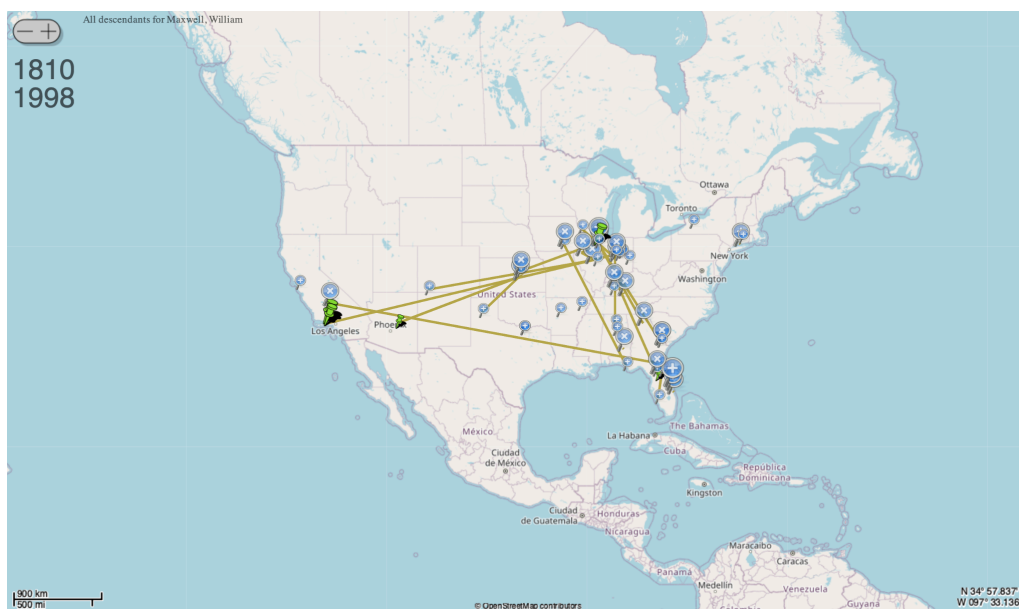


Figura 5. Geolocalización de una persona y toda su descendencia.

## 10.2 FamilySearch

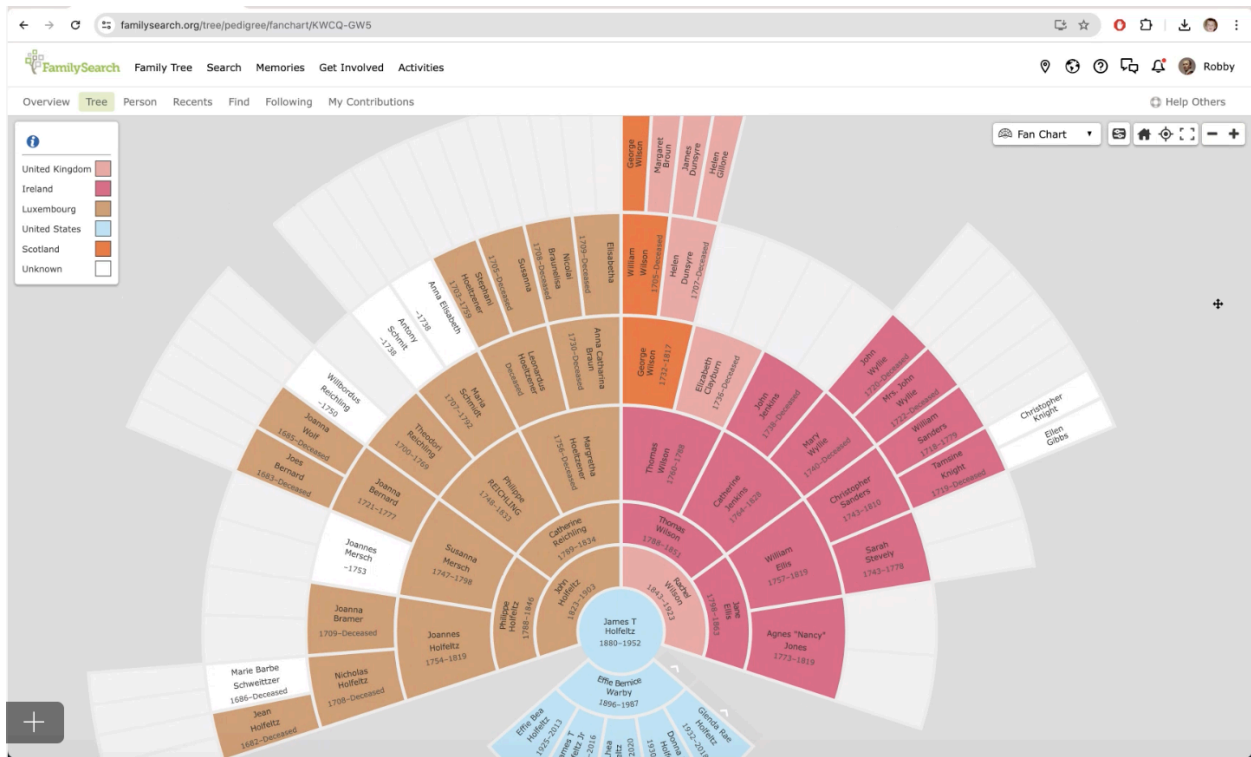


Figura 6. Vista de abanico

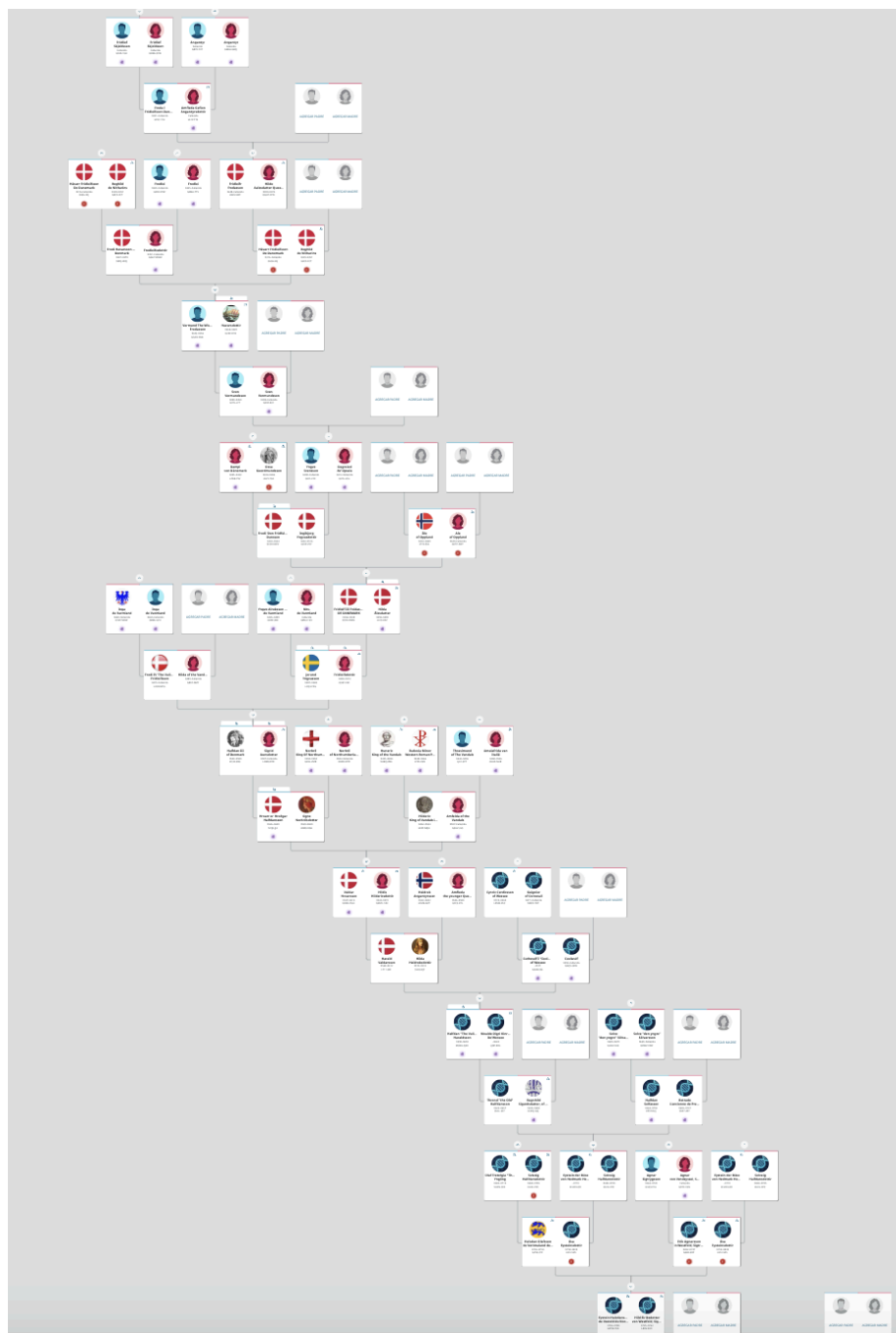


Figura 7. Vista de árbol.

## 11 Anexo: Arquitectura MyHeritage

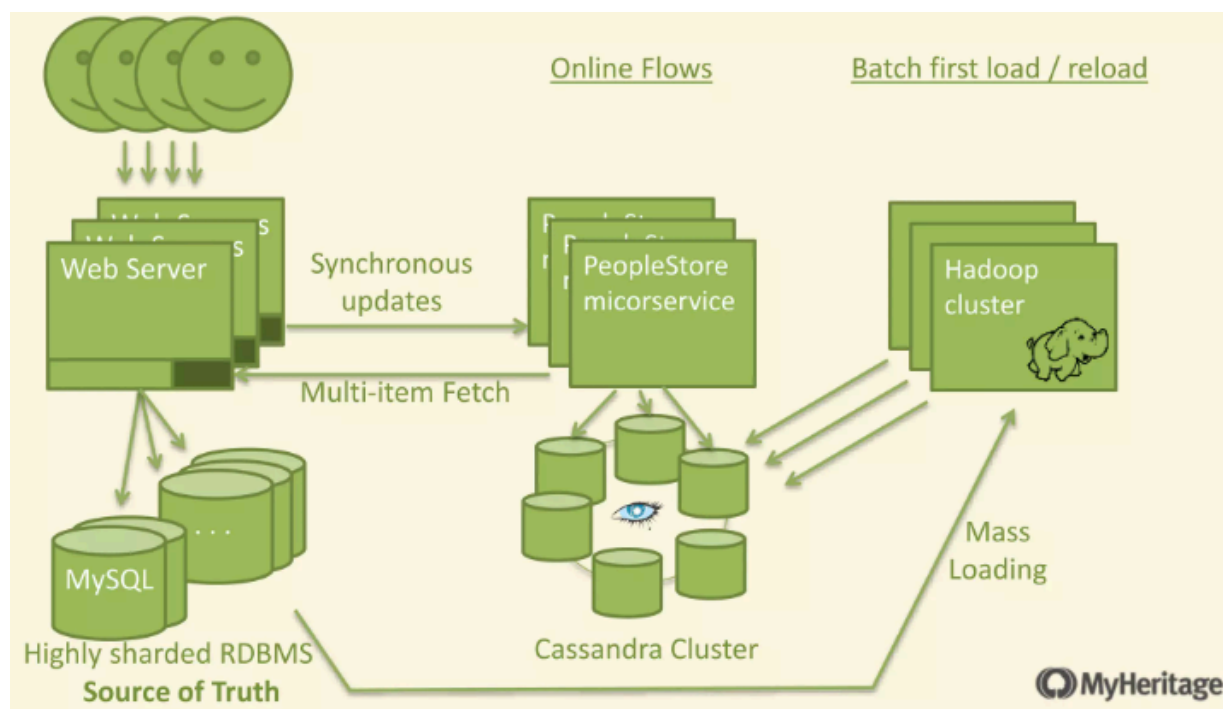


Figura 8. Arquitectura de datos utilizada por MyHeritage.