



Escuela Técnica Superior de  
**Ingeniería Informática**

## ANEXO A

# **Manual de instalación, despliegue y configuración de ATLAS Broadsea**

Realizado por  
**Da. María del Valle Alonso de Caso Ortiz**

Dirigido por  
Da. Silvia Rodríguez Mejías  
Dr. Carlos Parra Calderón

En el departamento de  
Innovación Tecnológica  
Hospital Universitario Virgen del Rocío

# Índice general

---

<b>1</b>	<b>Introducción y descripción de Broadsea . . . . .</b>	<b>1</b>
1.1.	Introducción al Manual . . . . .	1
1.2.	Descripción de Broadsea . . . . .	2
1.3.	Entorno Docker de Broadsea . . . . .	3
1.4.	Entorno PostgreSQL de Broadsea . . . . .	5
1.4.1.	La WebAPI . . . . .	6
1.4.2.	El Modelo común de datos (CDM) . . . . .	7
1.4.3.	Estructura de Broadsea . . . . .	8
1.5.	Recapitulación . . . . .	9
<b>2</b>	<b>Despliegue por defecto . . . . .</b>	<b>10</b>
2.1.	Requisitos para el despliegue . . . . .	10
2.2.	Deployment . . . . .	10
2.3.	Comprobación de despliegue correcto . . . . .	11
2.4.	Solución de posibles errores . . . . .	14
<b>3</b>	<b>Conexión local con la BD por defecto . . . . .</b>	<b>16</b>
3.1.	Requisitos para establecer la conexión . . . . .	16
3.2.	Deployment . . . . .	16
3.3.	Comprobación de conexión correcta . . . . .	18
3.4.	Solución de posibles problemas . . . . .	20
<b>4</b>	<b>Conexión local con BD externa . . . . .</b>	<b>22</b>
4.1.	Configuración manual . . . . .	22
4.1.1.	Requisitos previos . . . . .	22
4.1.2.	Deployment . . . . .	23
4.1.3.	Comprobación de conexión correcta . . . . .	26
4.1.4.	Solución de posibles problemas . . . . .	27
4.2.	Otras configuraciones avanzadas . . . . .	27
4.2.1.	Configurar la seguridad de Broadsea . . . . .	27
4.2.2.	Descargar otras BD . . . . .	28
<b>5</b>	<b>Configuración del Vocabulario . . . . .</b>	<b>29</b>
5.1.	Configuración manual . . . . .	29
5.1.1.	Requisitos previos . . . . .	29
5.1.2.	Deployment . . . . .	31
5.1.3.	Comprobación de configuración correcta . . . . .	32
5.1.4.	Solución de posibles errores . . . . .	34
5.2.	Otras configuraciones avanzadas . . . . .	34
5.2.1.	Configuración de Apache Solr . . . . .	35
5.2.2.	Configuración de PHOEBE . . . . .	35

<b>Bibliografía . . . . .</b>	<b>36</b>
-------------------------------	-----------

# 1. Introducción y descripción de Broadsea

---

## 1.1. Introducción al Manual

Este manual ha sido redactado entre Febrero y Marzo de 2024 con motivo de complementar como anexo al Trabajo de Fin de Grado *Extrayendo conocimiento a partir de análisis clínico de datos CDM usando la herramienta Atlas* de la misma autora, María del Valle Alonso de Caso Ortiz, y sobre la misma herramienta, ATLAS.

Dicha herramienta, por ser de código abierto, presenta multitud de información esparcida y repartida de forma pseudo-ordenada entre repositorios de github y foros de la comunidad OHDSI pero no se encontró ningún manual que recopilase toda la información de instalación y despliegue en un único sitio. Por tanto, a la hora de instalar y desplegar la herramienta, con objeto de su utilización en el TFG, se encontraron numerosas dificultades para encontrar procedimientos concretos y guiados.

Si bien había mucha información en la red sobre cómo realizar diferentes tipos de instalaciones, en la mayoría de los casos presentaban (i) información poco específica con asunción de conocimientos previos no mencionados (ii) información difícilmente accesible por presentarse en diferentes localizaciones en la red (iii) poca información de resolución de problemas, mayoritariamente presente en foros con problemas específicos de los cuales había que sacar conocimientos generales, entre otros.

De esta dificultad experimentada por la alumna fue de donde se originó la idea de redactar este anexo, para aportar a la comunidad OHDSI y a cualquier usuario que pretendiese utilizar la herramienta, una guía detallada paso a paso que facilitase y agrupase toda la información necesaria y la resolución de posibles conflictos durante la instalación, despliegue y configuración de ATLAS en el entorno de Broadsea.

No obstante, no es obvio el fin último del manual de acompañar al Trabajo Fin de Grado, desarrollado en compañía del grupo de Innovación Tecnológica del Hospital Universitario Virgen del Rocío, por lo que, frente a la multitud de posibilidades de configuración de distintos aspectos de la herramienta, aunque el manual presenta todos ellos, en numerosas ocasiones solo se centra en los procedimientos que aplican a las necesidades del TFG y del departamento del Hospital.

Es decir, el manual se desarrolla en un contexto de implementación de ATLAS Broadsea bajo los requisitos de una organización pública, el HUVR, con fines de ayudar y dar soporte a las investigaciones realizadas en el mismo, y bajo la supervisión continua de Da. Silvia Rodríguez Mejías y Dr. Carlos Parra Calderón.

Toda la información que se ha generado durante y tras la redacción de este

documento, y del TFG, se encuentra en un repositorio de github público, para permitir el acceso de cualquier usuario a comprobar archivos generados de variables, logs, scripts de código, etcétera. Para mayor información consultar el link del repositorio [1].

## 1.2. Descripción de Broadsea

Con objeto de la implementación de ATLAS satisfaciendo el marco del desarrollo del Trabajo Fin de Grado y el entorno de investigación del hospital (véase 1.1), se decide adoptar Broadsea como la alternativa más sencilla para implementar, desplegar y configurar la herramienta.

Broadsea es un proyecto basado en Docker que permite desplegar un entorno de herramientas, configuraciones y dependencias OHDSI de la manera más sencilla hasta el momento. De hecho, la misma organización la presenta, textualmente, como “*la forma más sencilla de instalar (y actualizar) las herramientas OHDSI*” [2].

Lo que comenzó, en su primera versión, como un simple contenedor que albergase imágenes de la WebAPI de ATLAS y RStudio [3] ha evolucionado hasta la tercera versión en la que Broadsea alberga la mayoría de herramientas OHDSI, creando un entorno virtual de desarrollo muy completo.

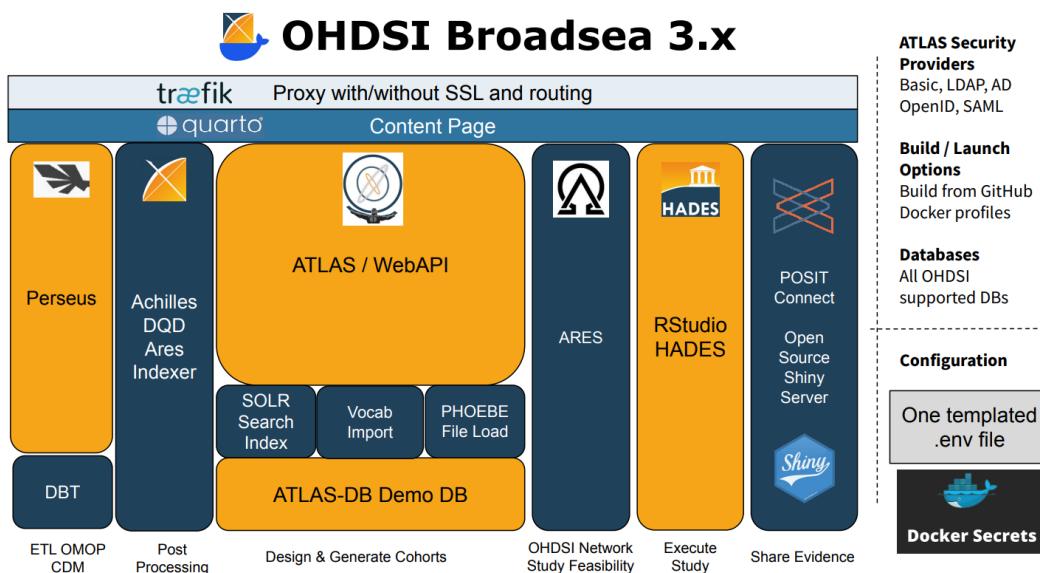


Figura 1.1: Vista general de todos los componentes de Broadsea. Extraída de [3].

Este manual tan solo se centra en la instalación, despliegue y configuración de ATLAS (y la WebAPI) pero las posibilidades de utilización de otras herramientas son múltiples. A partir de este momento se le denomina ATLAS Broadsea a la herramienta ATLAS que ofrece el entorno Broadsea.

## 1.3. Entorno Docker de Broadsea

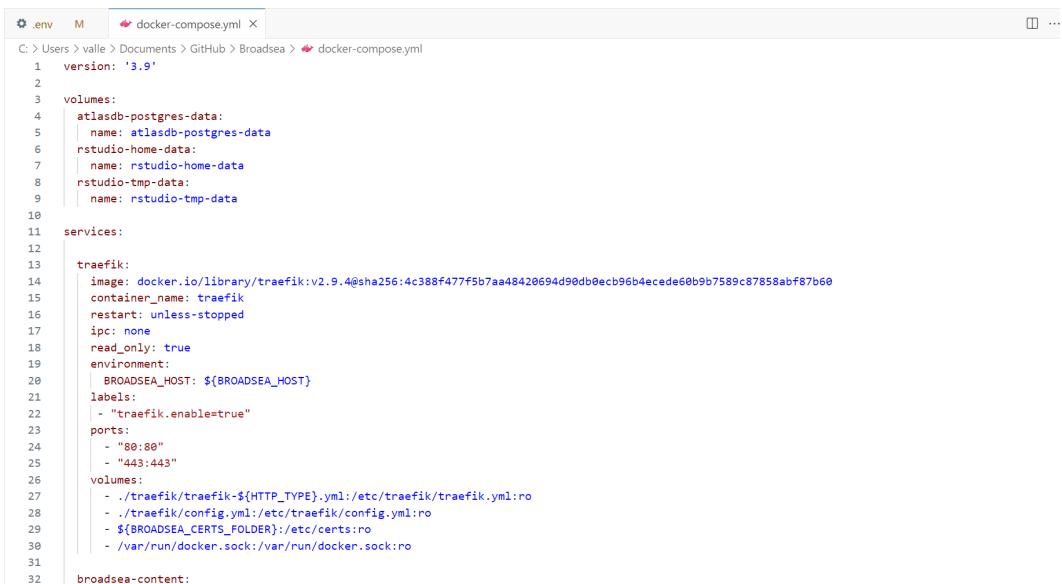
En la práctica, el entorno de Broadsea se despliega como un multicontenedor Docker con varios perfiles preconfigurados que permiten ir añadiendo o eliminando configuraciones avanzadas del entorno. La descripción general del multicontenedor y sus perfiles se encuentra online en el repositorio de github de Broadsea [4].

### Archivos de orquestación de contenedores

Existen dos archivos clave en el repositorio que orquestan y coordinan los parámetros de configuración del multicontenedor: el docker-compose.yml y el archivo .env.

1. El docker-compose.yml o simplemente docker-compose es el archivo docker que construye el entorno Broadsea, en él se describen los diferentes perfiles y sus parámetros y configuraciones más importantes. Es decir, cuando se construye el multicontenedor o se pretende ejecutar alguno de sus perfiles se llama a este archivo, que es el que contiene toda la información Docker relevante (véase 2.2).

Este archivo, por la relevancia que tiene en cuanto a la construcción del entorno Docker, no debería ser modificado, aunque puede ser de gran interés para comprobar la configuración o incluso la relación entre los diferentes perfiles, contenedores o volúmenes. A continuación se muestra una captura de pantalla a modo de ejemplo del archivo, abierto en Visual Studio Code.



```
C: > Users > vale > Documents > GitHub > Broadsea > docker-compose.yml
1  version: '3.9'
2
3  volumes:
4    | atlasdb-postgres-data:
5    |   name: atlasdb-postgres-data
6    | rstudio-home-data:
7    |   name: rstudio-home-data
8    | rstudio-tmp-data:
9    |   name: rstudio-tmp-data
10
11 services:
12
13   traefik:
14     image: docker.io/library/traefik:v2.9.4@sha256:4c388f477f5b7aa48420694d90db0ecb96b4ecede60b9b7589c87858abf87b60
15     container_name: traefik
16     restart: unless-stopped
17     ipc: none
18     read_only: true
19     environment:
20       BROADSEA_HOST: ${BROADSEA_HOST}
21     labels:
22       - "traefik.enable=true"
23     ports:
24       - "80:80"
25       - "443:443"
26     volumes:
27       - ./traefik/traefik-${HTTP_TYPE}.yml:/etc/traefik/traefik.yml:ro
28       - ./traefik/config.yml:/etc/traefik/config.yml:ro
29       - ${BROADSEA_CERTS_FOLDER}:/etc/certs:ro
30       - /var/run/docker.sock:/var/run/docker.sock:ro
31
32 broadsea-content:
```

Figura 1.2: Captura de pantalla de las primeras líneas del docker-compose.yml

2. El archivo .env es el archivo que contiene todas las configuraciones y parámetros modificables o customizables del entorno de Broadsea.

A diferencia del docker-compose, la modificación de parámetros de este archivo será requerida en numerosas ocasiones para activar o desactivar ciertos parámetros o modificar algunas configuraciones del entorno. No obstante, también puede ser de gran interés para comprobar información relevante sobre los parámetros de funcionamiento del multicontenedor. A continuación se muestra una captura de pantalla a modo de ejemplo del archivo, abierto en Visual Studio Code.

```

C: > Users > value > Documents > GitHub > Broadsea > .env
1 #####
2 # Section 1:
3 # Broadsea Host
4 #####
5 DOCKER_ARCH="linux/amd64" # change this to linux/arm64 if using Mac Silicon, otherwise keep as-is
6 BROADSEA_HOST="127.0.0.1" # change to your host URL (without the http part)
7 HTTP_TYPE="http" # if using https, you need to add the crt and key files to the ./certs folder
8 BROADSEA_CERTS_FOLDER="./certs"
9 #####
10 #####
11 # Section 2:
12 # Atlas GUI configuration
13 #####
14 #####
15 ATLAS_INSTANCE_NAME="Broadsea"
16 ATLAS_COHORT_COMPARISON_RESULTS_ENABLED="false"
17 ATLAS_USER_AUTH_ENABLED="false" # set to true if using security, but ensure you fill out the WebAPI/Atlas security sections below
18 ATLAS_PLP_RESULTS_ENABLED="false"
19 #####
20 #####
21 # Section 3:
22 # WebAPI Database configuration
23 #####
24 #####

```

**Figura 1.3:** Captura de pantalla de las primeras líneas del archivo .env.

Como se puede apreciar en la imagen, este archivo está organizado en diferentes secciones, y presenta numerosos comentarios para facilitar al usuario la comprensión de los parámetros que debe modificar para establecer las configuraciones avanzadas del entorno.

Hay doce secciones distintas: (1) Contiene la configuración del servidor donde se aloja Broadsea, (2) contiene la configuración de la interfaz gráfica de ATLAS, (3) contiene la configuración de la base de datos de ATLAS, (4) contiene la configuración del protocolo de seguridad de ATLAS, (5) contiene la configuración del protocolo de seguridad de la WebAPI, (6) contiene la configuración de ATLAS y la WebAPI mediante Github, (7) contiene la configuración del vocabulario de Apache Solr, (8) contiene la configuración de identificación en HADES, (9) contiene la configuración del vocabulario de Postgres, (10) contiene la configuración de PHOEBE, (11) contiene la configuración de ARES, (12) contiene la configuración de la página inicial de Broadsea.

## Perfiles de configuración

En cuanto a los perfiles, a la hora de desplegar por primera vez el multicontenedor (véase 2), se ejecuta el perfil default, es decir, por defecto, que instala seis contenedores que proveen las herramientas básicas de Broadsea: la WebAPI, el traefik, ATLAS y HADES y la base de datos de ATLAS (véase Figura 1.1). A partir de este momento, el resto de configuraciones adicionales se realizan a través de los once perfiles restantes, descritos en el docker-compose y en el repositorio [4]

## Volúmenes de almacenamiento de datos

Además de los contenedores, Broadsea implementa tres volúmenes por defecto: un volumen para albergar la base de datos de ATLAS (`atlasdb-postgres-data`) y dos para la información de HADES (`rstudio-home-data` y `rstudio-tmp-data`).

Los volúmenes de docker proporcionan una forma de persistir datos más allá del ciclo de vida de un contenedor y permiten compartir datos entre contenedores o entre un contenedor y el host. La información sobre los volúmenes y las relaciones que guardan con los contenedores se encuentra en el `docker-compose`. Una vez que se despliegue Broadsea en el equipo, hay dos formas de comprobar qué volúmenes se están ejecutando: (i) a través de la interfaz gráfica de Docker Desktop - Figura 1.4 (ii) a través del cmd - Figura 1.5.

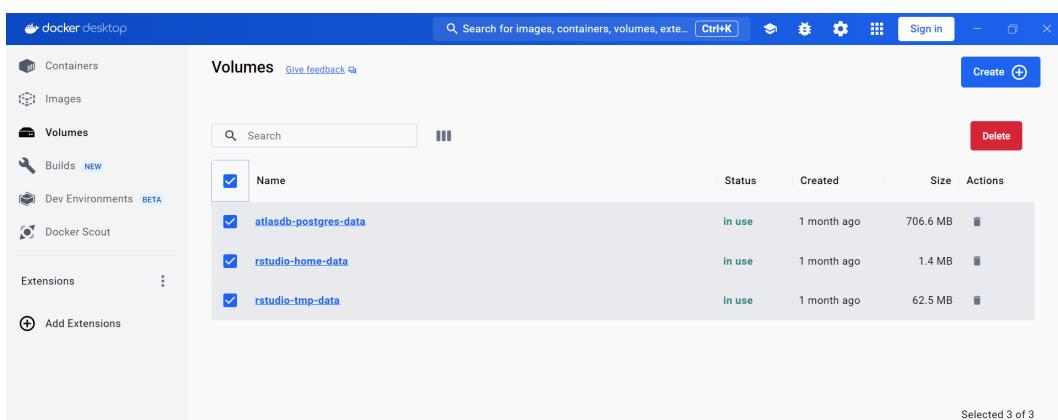


Figura 1.4: Captura de pantalla del panel volumes de Docker Desktop

```
C:\Users\valle>docker volume ls
DRIVER      VOLUME NAME
local      atlasdb-postgres-data
local      rstudio-home-data
local      rstudio-tmp-data
```

Figura 1.5: Captura de pantalla del comando `docker volume ls` en el cmd

El volumen (`atlasdb-postgres-data`) será de especial interés durante la configuración de la base de datos de ATLAS (véase 3).

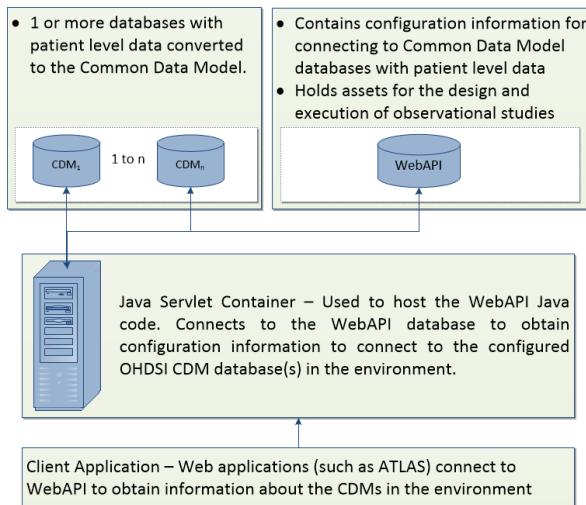
## 1.4. Entorno PostgreSQL de Broadsea

La configuración y relación entre las diferentes bases de datos que interaccionen con Broadsea se realiza a través de la WebAPI de ATLAS, que se aloja en un servidor de base de datos PostgreSQL. La documentación sobre la WebAPI es muy extensa y se encuentra en un repositorio principal de github [5] que se divide en diferentes secciones de interés.

### 1.4.1. La WebAPI

#### Estructura teórica

En la wiki del repositorio [6] se da información más detallada sobre la webAPI y se define como "una aplicación basada en Java diseñada para proporcionar un conjunto de servicios web RESTful para interactuar con una o más bases de datos" y se caracteriza según la siguiente estructura:



**Figura 1.6:** Estructura de la WebAPI. Extraída de [6]

En cuanto a esta estructura es importante anotar dos observaciones cruciales para el correcto funcionamiento de la WebAPI:

1. La base de datos propia de la WebAPI, que almacena toda la información relevante de su propia configuración, solo puede alojarse en una base de datos PostgreSQL. Por tanto, Broadsea solo puede desplegarse, necesariamente sobre un entorno PostgreSQL.
2. Las bases de datos externas que se conecten a la WebAPI (véase 4), pueden alojarse en cualquier tipo de base de datos pero deben estar correctamente normalizadas al modelo común de datos (CDM) de OMOP.

#### Estructura en Postgre

Es muy importante conocer el papel de la WebAPI y su funcionamiento en PostgreSQL para el despliegue de todo el entorno de Broadsea. La documentación sobre la configuración de los esquemas de la WebAPI y el CDM puede consultarse en su repositorio de github [7].

La WebAPI se implementa en Postgre a través del esquema `webapi`, que contiene entre otras muchas, dos tablas de especial relevancia:

- sources: Esta tabla es la que registra e integra las fuentes, es decir, las bases de datos externas, que se conectan a la WebAPI, según lo visto en la Figura 1.6.
- sources\_daimon: Esta tabla es la que registra la estructura de esquemas de cada fuente, y garantiza que se cumpla la estructura del modelo común de datos de OMOP, según lo visto en la Figura 1.6.

Opcionalmente también puede añadirse el esquema `temp` para permitir a la WebAPI almacenar información de forma temporal y el esquema `webapi_security` si se quiere configurar el protocolo de seguridad de Broadsea (véase 4.2.1).

### 1.4.2. El Modelo común de datos (CDM)

#### Estructura teórica

La explicación en profundidad del modelo común de datos de OMOP es muy extensa y no concierne directamente a los contenidos que desarrolla este manual aunque sí es importante conocer, al menos brevemente, su estructura teórica y su implementación en Postgre. La documentación e información extensa sobre el CDM se encuentra en su repositorio de github [8], en github pages [9] y en el Libro de OHDSI [10].

Recientemente OHDSI lanzó la sexta versión del CDM aunque aún no está totalmente integrada con todas las herramientas, por lo que el manual utiliza en la implementación de Broadsea la quinta versión del CDM. Esta versión, a modo visual, presenta la siguiente estructura:

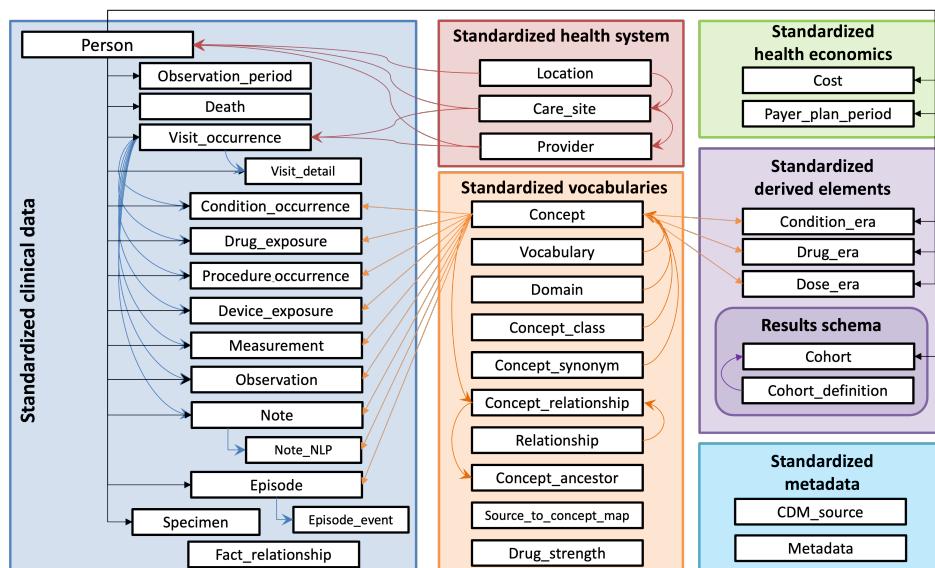


Figura 1.7: Estructura del CDM v5.4. Extraída de [9]

Es importante, realizar una observación relevante sobre la estructura del CDM, que es la interacción entre el modelo de datos con el Vocabulario. El vocabulario se

representa como una tabla más en la estructura pero es una entidad externa muy extensa y con configuración propia. La documentación sobre el Vocabulario se encuentra en la wiki de su repositorio de github [11] y en el Libro de OHDSI [10].

### Estructura en Postgre

Para interactuar correctamente con Broadsea, las bases de datos externas que se integren con la WebAPI deben estar estructuradas según el CDM de OMOP. La configuración e implementación del modelo de datos se describe en su repositorio de github [7]. En resumen, una base de datos normalizada a OMOP debe contener al menos los siguientes esquemas:

- **cdm**: Es el esquema que contiene todas las tablas con la información clínica de la base de datos estructurada según el CDM (véase Figura 1.7). Su presencia es obligatoria.
- **results**: Es el esquema donde se almacenan los resultados obtenidos tras los análisis que realiza la herramienta. Se separa del **cdm** para evitar mezclar la información de la base de datos de solo lectura y la información producida por el usuario. Su presencia es obligatoria.
- **vocab**: Es el esquema donde se almacena el Vocabulario. Su configuración es obligatoria aunque puede aparecer implícito en el **cdm** o como un esquema externo. Si hay varias bases de datos que utilicen el mismo Vocabulario no es necesario configurar un Vocabulario para cada una, basta con que se implemente uno y el resto apunten a él.

#### 1.4.3. Estructura de Broadsea

Por último, presentar concretamente la estructura que adopta Broadsea al desplegarse en Postgre. Pues si bien Broadsea se virtualiza gracias a Docker, se puede acceder a su base de datos de forma local gracias a la conexión con el servidor Postgre de Broadsea. La información relevante a la base de datos de Broadsea se puede consultar en su repositorio de github [12].

En su configuración por defecto (véase 3), el servidor de Broadsea se configura en el localhost (bien en el 0.0.0.0 o en 127.0.0.1), en el puerto 5432 [12] y alberga una única base de datos en la que combina la información de la WebAPI y una pequeña muestra de la base de datos de Eunomia. Para acceder a la documentación sobre Eunomia se consultar su repositorio de github [13] o github pages [14].

Es decir, la base de datos inicial de Broadsea presenta cinco esquemas:

- **public**: Es el esquema por defecto que se crea con cualquier base de datos Postgre.
- **webapi**: Corresponde al esquema webapi de la estructura de la WebAPI. Presenta la información de configuración del entorno.

- `webapi_security`: Es un esquema opcional para activar la seguridad de la WebAPI y ATLAS. No se utiliza en el manual.
- `demo_cdm`: Corresponde al esquema `cdm` de la estructura del CDM. Alberga el contenido de la base de datos de Eunomia y una pequeña muestra implícita del Vocabulario.
- `demo_cdm_results`: Es el esquema que corresponde al esquema `results` de la estructura del CDM.

Es importante anotar que la configuración por defecto de Broadsea trae una pequeña muestra del Vocabulario pero muy insignificante, por lo que para realizar análisis en profundidad será necesario configurar un esquema que albergue información más extensa (véase [5](#)).

## 1.5. Recapitulación

A esta altura, queda recogido en [1.1](#) la introducción al manual, el motivo por el que se redacta, la necesidad que subyace al mismo y el contexto en el que se realiza la instalación y en [1.2](#), [1.3](#), [1.4](#), la descripción teórica de Broadsea y el desglose en las distintas herramientas que permiten su despliegue.

Por tanto, a partir de este momento comienza en [2](#), [3](#), [4](#) y [5](#) la implementación práctica de todos los contenidos presentados, orientados específicamente a la instalación y configuración de la herramienta ATLAS.

En conclusión, si bien hasta ahora se presentaba Broadsea de forma general, el enfoque de las siguientes secciones gira concretamente entorno a ATLAS Broadsea.

## 2. Despliegue por defecto.

---

El despliegue de ATLAS en Docker es muy sencillo y está muy bien documentado en su repositorio de github [4]. Igualmente, en este manual se detallan nuevamente los pasos para la configuración y despliegue de la herramienta, añadiendo algunos pasos relevantes adicionales.

### 2.1. Requisitos para el despliegue

1. Descargar e instalar Docker. Lo más sencillo es seguir las instrucciones de la [página web oficial](#) para la descarga y seguir la configuración por defecto para la instalación.
2. Descargar e instalar Git. Lo más sencillo es seguir las instrucciones de la [página web oficial](#) para la descarga y seguir la configuración por defecto para la instalación.

### 2.2. Deployment

En este primer despliegue rápido de ATLAS, se desplegarán las configuraciones por defecto de la herramienta, siguiendo la guía de implementación rápida (*Quick Start*) del mismo repositorio de github.

1. Por tanto, el primer paso para desplegar ATLAS es clonar localmente el repositorio de github de Broadsea. Una forma rápida de hacerlo es copiar la siguiente línea de código en el cdm.

---

```
1 git clone https://github.com/OHDSI/Broadsea.git
```

---

2. El segundo paso, es desplegar el contenedor docker. Para ello, situar el puntero del cdm, en la carpeta donde se ha copiado el repositorio de github de Broadsea.

---

```
1 cd ruta\del\repositorio\Broadsea\local
```

---

Una vez situado en la carpeta raíz del repositorio, se ejecuta el comando que instalará y desplegará el perfil por defecto del contenedor docker en la máquina local.

---

```
1 docker compose pull && docker-compose --profile default up -d
```

---

## 2.3. Comprobación de despliegue correcto

Se puede comprobar que se ha instalado correctamente el contenedor de Broadsea en la máquina local de distintas formas, tal y como se presenta a continuación.

1. La forma más sencilla de interactuar con el contenedor de Broadsea es a través de Docker Desktop. Ejecutando dicho programa, en la sección "containers" se muestran todos los contenedores que están corriendo en el equipo. En este caso, debe aparecer un multi-contenedor llamado "broadsea" que contenga seis contenedores, tal y como se muestra en la Figura 2.1.

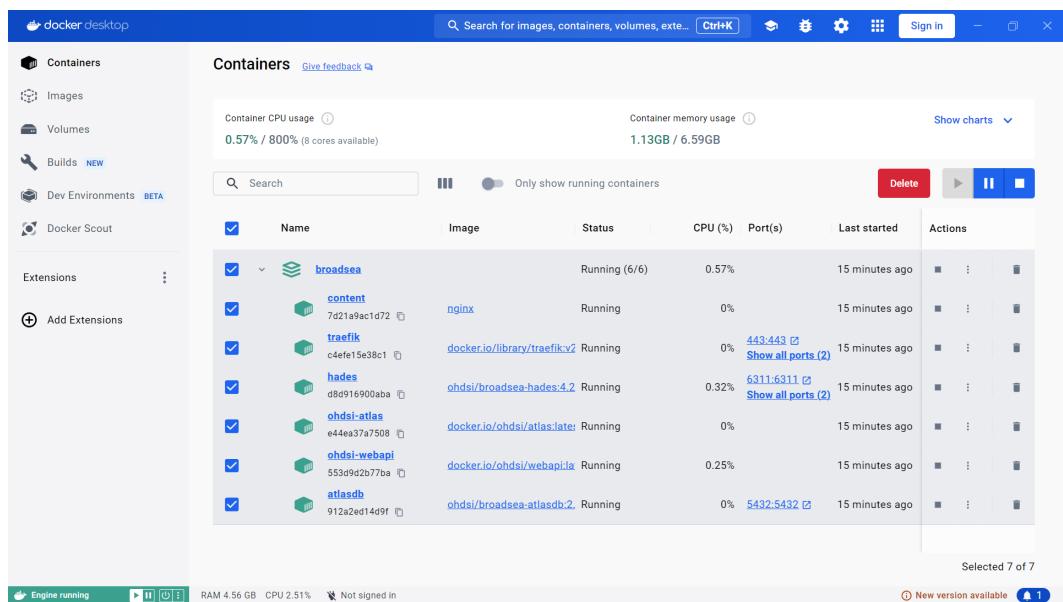


Figura 2.1: Captura de pantalla del contenedor Broadsea en Docker Desktop

Mediante el panel de control de Docker se puede iniciar, pausar o detener cada contenedor (o todos a la vez) fácilmente y en cualquier momento. Por esto se dice que Broadsea ofrece servicios *a-la-carte*.

2. Otra forma de interactuar con el contenedor Docker es a través del cmd, ejecutando el comando docker ps, que muestra un listado de todos los contenedores que está ejecutando la máquina local. Con esta estrategia deberían mostrarse igualmente los mismos seis contenedores pertenecientes a broadsea, tal y como se muestra en la Figura 2.2

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
7d21a9ac1d72	nginx	"sh /tmp/envsubst.sh" broadsea-content	6 weeks ago	Up 23 minutes	80/tcp
c4efef15e38c1	traefik:v2.9.4	"/entrypoint.sh traef..."	6 weeks ago	Up 23 minutes	0.0.0.0:8
0->80/tcp, 0.0.0.0:443->443/tcp		traefik			
d8d916900aba	ohdsi/broadsea-hades:4.2.1	"/usr/bin/supervisor..."	6 weeks ago	Up 23 minutes	0.0.0.0:6
311->6311/tcp, 0.0.0.0:8787->8787/tcp	broadsea-hades				
e44ea37a7508	ohdsi/atlas:latest	"sh /tmp/envsubst.sh"	6 weeks ago	Up 23 minutes	8080/tcp
		ohdsi-atlas			
553d9d2b77ba	ohdsi/webapi:latest	"/bin/sh -c 'exec ja..."	6 weeks ago	Up 23 minutes	8080/tcp
		ohdsi-webapi			
912a2ed14d9f	ohdsi/broadsea-atlasdb:2.0.0	"docker-entrypoint.s..."	6 weeks ago	Up 23 minutes (healthy)	0.0.0.0:5
432->5432/tcp	broadsea-atlasdb				

Figura 2.2: Captura de pantalla del comando docker ps en el cmd

3. Por último, para acceder a los servicios de Broadsea hay que abrir en el navegador web (Chrome recomendado) el servidor en el que se alojan los servicios. Por defecto, Broadsea se aloja en el servidor 127.0.0.1 (véase 1.4). Por tanto, introducir la dirección en el navegador para explorar las herramientas del contenedor, tal y como se muestra en la Figura 2.3.

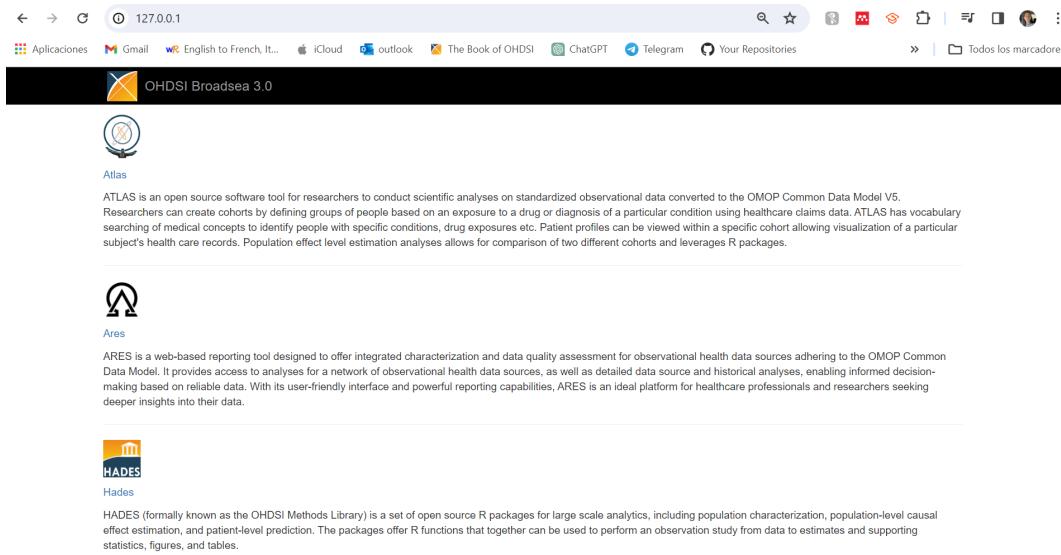


Figura 2.3: Captura de pantalla del servidor Broadsea ejecutado en Chrome

Se puede comprobar o modificar el servidor exacto dónde se aloja el contenedor revisando el parámetro BROADSEA\_HOST de la sección 1 del archivo .env (para mayor información véase 1.3) de la ruta local del repositorio clonado.

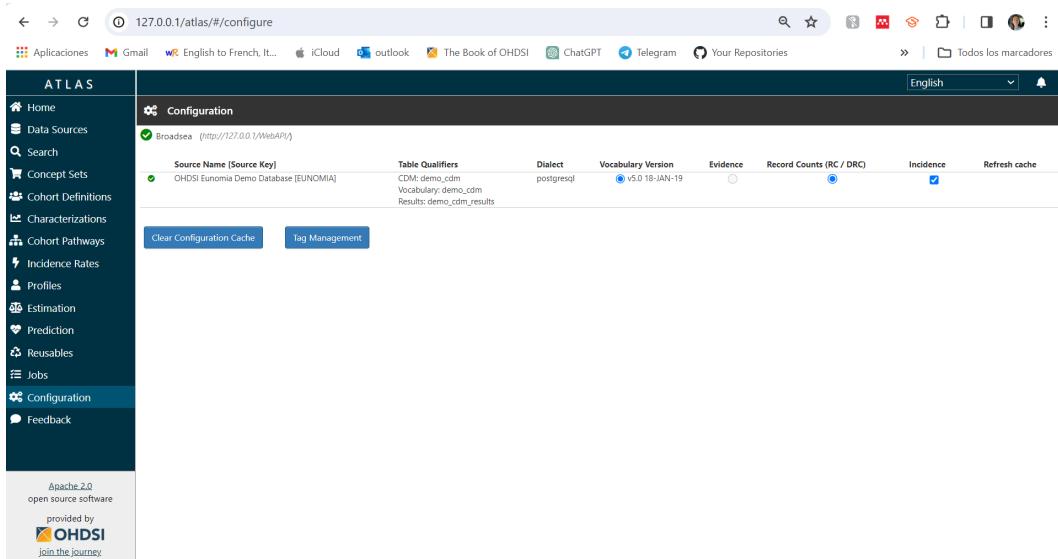
```
C:\> Users > valle > Documents > GitHub > Broadsea > env
1 ######
2 # Section 1:
3 # Broadsea Host
4 #####
5 DOCKER_ARCH="linux/amd64" # change this to linux/arm64 if using Mac Silicon, otherwise keep as-is
6 BROADSEA_HOST="127.0.0.1" # change to your host URL (without the http part)
7 HTTP_TYPE="http" # if using https, you need to add the crt and key files to the ./certs folder
8 BROADSEA_CERTS_FOLDER="./certs"
9
```

Figura 2.4: Captura de pantalla de la sección 1 del archivo .env.

En esta guía de implementación, no se va a modificar la dirección del servidor para la configuración de Broadsea, para no desconfigurar el entorno por defecto de la herramienta. Por tanto, el servidor de Broadsea durante todo el transcurso del manual será 127.0.0.1, tal y como se recomienda en [12] y como se muestra en la Figura 2.3

Es interesante notar que Broadsea permite el acceso interactivo a la herramienta Atlas, que es la que nos interesa en este caso, pero también a Ares y a Hades, otras dos herramientas muy relacionadas (véase 1.2).

La ejecución de ATLAS en Broadsea es similar a ATLAS demo [15], aunque con algunas diferencias. En primer lugar, Broadsea solo ejecuta, por defecto, una base de datos, que es la base de datos de Eunomia (véase 1.4).



**Figura 2.5:** Captura de pantalla de base de datos que utiliza ATLAS Broadsea

Por otra parte, y en contraste con la versión demo, ya no aparecen las entradas y estructuras que generan otros usuarios. La herramienta se presenta vacía, para ser completada solo con la información que el usuario local introduzca.

**Figura 2.6:** Captura de pantalla señalando el número de entradas de definición de cohorte que almacena ATLAS Broadsea

## 2.4. Solución de posibles errores

**Error: No se puede acceder a este sitio web. La página 127.0.0.1 ha rechazado la conexión.**

Problema: Al introducir el servidor de Broadsea en el buscador del navegador aparece esta página de error que impide conectar con el entorno.

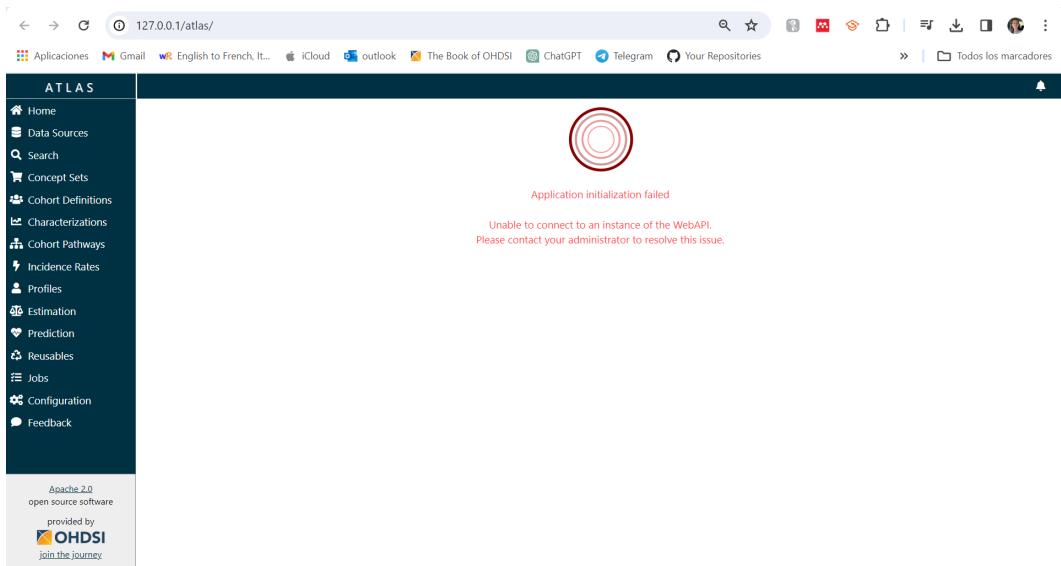
**Figura 2.7:** Captura de pantalla del error

Motivo: No se puede acceder al entorno Broadsea porque los contenedores Docker están detenidos.

Solución: Comprobar que todos los contenedores de docker implicados están corriendo. Para ello acceder a Docker Desktop y ejecutar todos los contenedores del multicontenedor de Broadsea. Una vez ejecutados todos los contenedores, si el problema persiste recargar varias veces la página.

**Error: Application initialization failed. Unable to connect to an instance of the WebAPI. Please contact your administrator to resolve this issue.**

Problema: Al abrir la herramienta ATLAS Broadsea aparece en rojo este error que impide conectar con la herramienta.



**Figura 2.8:** Captura de pantalla del error

Motivo: La aplicación falla al inicializarse porque no se está ejecutando correctamente el entorno docker de Broadsea.

Solución: Comprobar que todos los contenedores de docker implicados están corriendo. Para ello acceder a Docker Desktop y ejecutar todos los contenedores del multicontenedor de Broadsea. Una vez ejecutados todos los contenedores, si el problema persiste recargar varias veces la página.

# 3. Conexión local con la BD por defecto

---

Generalmente, resulta de interés acceder a la base de datos remota de Broadsea desde un administrador de bases de datos local. Docker almacena la base de datos de Broadsea en el volumen `atlasdb-postgres-data` (véase [1.3](#)) y crea un entorno de interoperabilidad e interconectividad a través de la WebAPI, que se aloja en un servidor Postgre (véase [1.4](#)).

Por tanto, para acceder a la base de datos de ATLAS Broadsea se requiere acceder al servidor remoto de Broadsea, alojado en el 127.0.0.1 (véase [2.2](#)), desde la instancia local de Postgres. La información relevante para esta implementación se encuentra en los siguientes repositorios de github [\[4\]](#) y [\[12\]](#) y en el foro de OHDSI "Question about Broadsea default database" [\[16\]](#).

## 3.1. Requisitos para establecer la conexión

1. Descargar e instalar la base de datos PostgreSQL. Lo más sencillo es seguir las instrucciones de la [página web oficial](#) para la descarga y seguir la configuración por defecto para la instalación.
2. Descargar e instalar una administrador de base de datos PostgreSQL. Se recomienda utilizar el administrador pgAdmin. Para ello, lo más sencillo es seguir las instrucciones de la [página web oficial](#) para la descarga y seguir la configuración por defecto para la instalación.

## 3.2. Deployment

Para establecer la conexión con la base de datos que utiliza ATLAS Broadsea, primero se debe establecer la conexión con la WebAPI. Se recomienda seguir las siguientes instrucciones:

1. En primer lugar, comprobar los parámetros de configuración de la WebAPI. Para ello, consultar la sección 3 del archivo `.env`.

```

.env M X docker-compose.yml
C: > Users > valle > Documents > GitHub > Broadsea > .env
20 #####
21 # Section 3:
22 # WebAPI Database configuration
23 #####
24
25 # Keep as-is if using Broadsea to launch the WebAPI postgres, replace if using an external postgres instance
26 WEBAPI_DATABASE_URL="jdbc:postgresql://broadsea-atlasdb:5432/postgres"
27 WEBAPI_DATABASE_USERNAME="postgres"
28 WEBAPI_DATABASE_PASSWORD="mypass"
29 WEBAPI_DATABASE_SCHEMA="webapi"
30

```

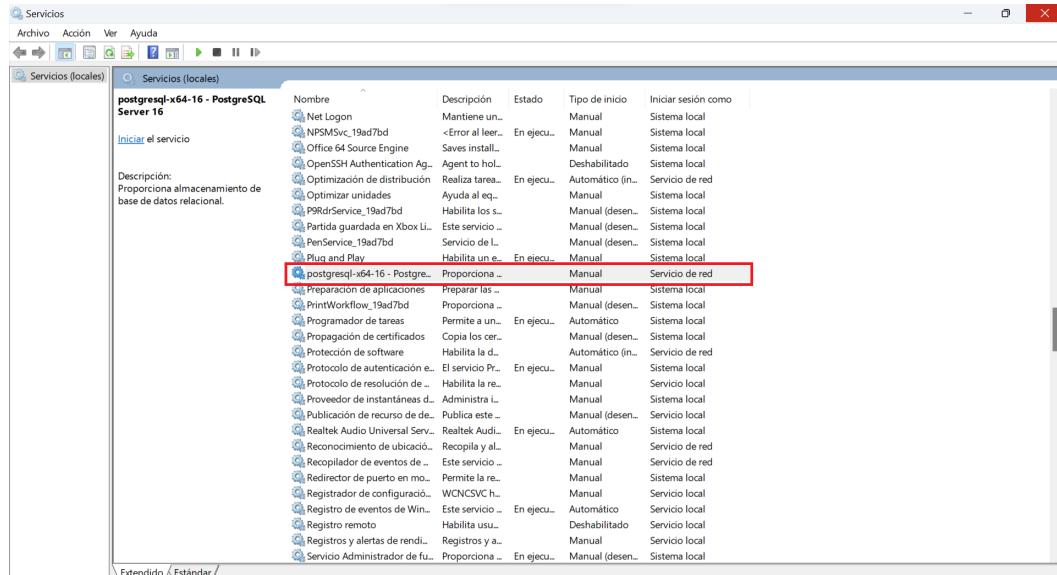
**Figura 3.1:** Captura de pantalla de la sección 3 del archivo .env.

Tal y como se muestra en la Figura 3.1, en el archivo por defecto, se presenta la contraseña para acceder a la WebAPI (*password=mypass*) y el puerto que utiliza (*port=5432*).

2. El servidor por defecto de Broadsea se solapa con el servidor local por defecto de PostgreSQL puesto que ambos se alojan en la dirección 127.0.0.1 y en el puerto 5432.

Por tanto, para evitar este solapamiento existen varias estrategias. Una estrategia podría ser cambiar el puerto de la instancia local de Postgre para que cada instancia se aloje en espacios distintos sin solapamiento. Sin embargo, en este caso, directamente se ha detenido el servicio local de PostgreSQL, de forma que el servidor quede totalmente libre para albergar la base de datos de Broadsea.

Para detener el servicio local de PostgreSQL lo más sencillo es abrir la aplicación *servicios* buscar el servidor de poste y deterlo, tal y como se muestra en la Figura 3.2.



**Figura 3.2:** Captura de pantalla de la aplicación de servicios.

3. El último paso consiste en registrar el servidor a través del administrador de base de datos instalado en el equipo, en este caso pgAdmin.

Una vez que se haya liberado el puerto 5432, se puede registrar un nuevo servidor en dicho puerto que acceda a la base de datos de Broadsea. Para ello, abrir el administrador de la base de datos PostgreSQL, pgAdmin 4, y registrar un nuevo servidor. Los parámetros de configuración de este nuevo servidor deben ser los mismos que se establecen en la sección 3 del archivo `.env`. Los parámetros fundamentales son:

---

```

1 host = 127.0.0.1
2 port = 5432
3 user = postgres
4 password = mypass

```

---

4. Tras registrar el servidor correctamente, debe aparecer una base de datos con cinco esquemas: `demo_cdm`, `demo_cdm_results`, `public`, `webapi`, `webapi_security` (véase [1.4](#)).

### 3.3. Comprobación de conexión correcta

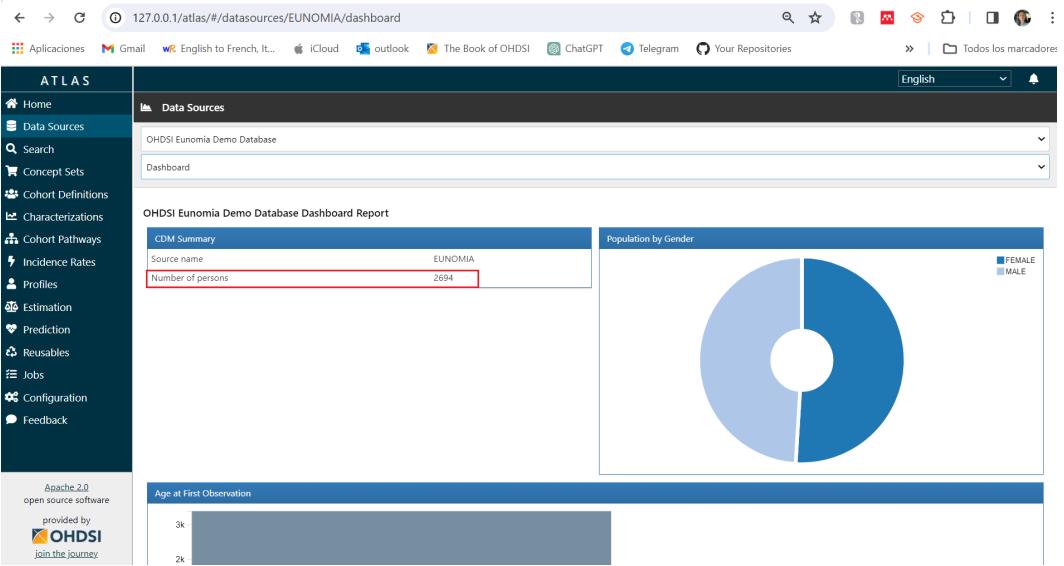
Una vez se ha registrado el servidor de Broadsea a través de pgAdmin es importante comprobar que la conexión se ha realizado correctamente, sin pérdida de información. Para ello se pueden realizar distintas comprobaciones:

1. Por ejemplo, comprobar el número de filas que recupera pgAdmin de la tabla `person` del esquema `demo_cdm`, tal y como se muestra en la Figura [3.3](#).

person_id	gender_concept_id	year_of_birth	month_of_birth	day_of_birth	birth_datetime	race_concept_id	ethnicity_concept_id
1	1	8507	1949	1	27 1949-01-27 00:00:00	8527	
2	2	8532	1920	6	1 1920-06-01 00:00:00	8527	
3	3	8507	1916	1	3 1916-01-03 00:00:00	8527	
4	5	8507	1968	7	24 1968-07-24 00:00:00	8515	
5	6	8532	1963	12	31 1963-12-31 00:00:00	8516	
6	7	8532	1968	11	6 1968-11-06 00:00:00	0	380
7	9	8532	1978	7	20 1978-07-20 00:00:00	8516	
8	11	8507	1953	2	3 1953-02-03 00:00:00	8527	
9	12	8532	1963	1	30 1963-01-30 00:00:00	8527	
10	16	8532	1971	10	13 1971-10-13 00:00:00	8527	
11	17	8532	1950	12	11 1950-12-11 00:00:00	8527	
12	18	8532	1965	11	17 1965-11-17 00:00:00	8527	
13	19	8532	1949	1	30 1949-01-30 00:00:00	0	380

**Figura 3.3:** Captura de pantalla de pgAdmin.

El número de filas que recupere pgAdmin debe ser igual al número de personas que muestra ATLAS en la sección Data Source/Dashboard, en este caso son 2694 personas (Figura [3.4](#)).



**Figura 3.4:** Captura de pantalla de ATLAS Data Sources/Dashboard.

2. Otra forma de comprobar que la conexión es correcta y una forma alternativa de realizarla, con el fin de detectar posibles problemas durante la implementación, es ejecutar el siguiente script de código en R, que realiza la conexión con la base de datos a través de RStudio:

La sección central derecha muestra el explorador de proyectos 'Project: (None)' con una conexión a 'DatabaseConnector (postgres)' que enumera los esquemas 'public', 'information\_schema', 'pg\_catalog', 'demo\_cdm', 'demo\_cdm\_results' y 'webapi'. La sección inferior izquierda es la consola 'Console' que muestra la ejecución del script y la conexión exitosa: 

```

> #Conexión con la BD
> connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = "postgres", user = "postgres",
+ password = "mypass", server = "127.0.0.1/postgres",
+ port = 5432,
+ pathToDriver = "C:/Users/valle/Documents/PostgreSQL")
> connection <- DatabaseConnector::connect(connectionDetails = connectionDetails)
Connecting using PostgreSQL driver
> 
```

La sección inferior derecha es la barra de recursos 'R Resources' que incluye enlaces a 'Learning R Online', 'CRAN Task Views', 'R on StackOverflow' y 'Getting Help with R'."/>

**Figura 3.5:** Captura de pantalla de script de RStudio.

El código que se muestra en la Figura está extraído (modificado) del foro [16] y está disponible en el repositorio de github del TFG en la ruta environment/anexes/manual/rscript [1].

## 3.4. Solución de posibles problemas

### Al registrar el servidor se queda pillado en "Saving"

Problema: A la hora de registrar el servidor por primera vez a través de pgAdmin, pulsar la opción "save", se mantiene cargando "saving" eternamente.

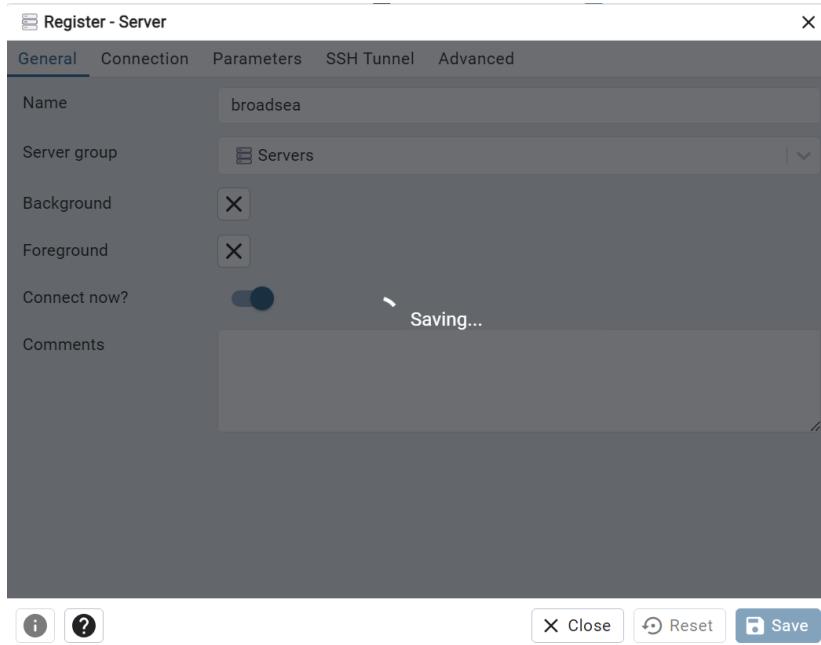


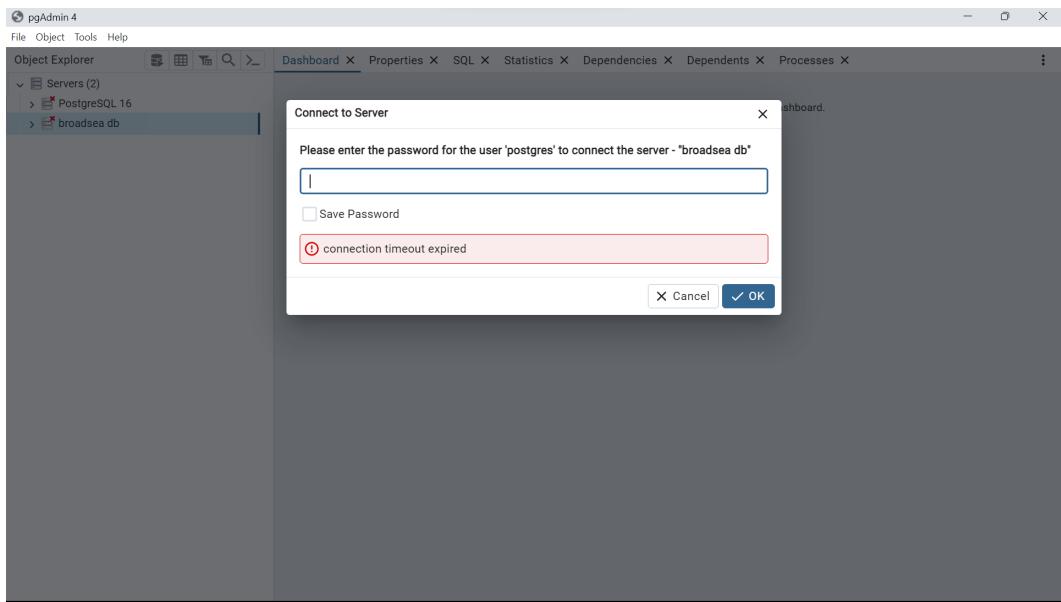
Figura 3.6: Captura de pantalla de error.

Motivo: No permite registrar el servidor, probablemente porque el puerto que se pretende utilizar ya está en uso.

Solución: Tal y como se sugiere en [3.2](#) o bien cambiar el puerto que utiliza la instancia local de Postgre o bien directamente detener todo el servicio local de Postgre a través de la aplicación "Servicios".

### Error: Connection timeout expired

Problema: Al acceder al servidor de Broadsea e introducir la contraseña correctamente aparece dicho error que impide acceder al servidor.



**Figura 3.7:** Captura de pantalla de error.

Motivo: El tiempo de conexión con el servidor espira porque no se puede acceder a él porque el contenedor Docker que alberga la base de datos de Broadsea no está ejecutándose.

Solución: Comprobar que todos los contenedores de docker implicados están corriendo, especialmente el contenedor atlasdb. Para ello acceder a Docker Desktop y ejecutar los contenedores implicados o directamente todos los pertenecientes al multicontenedor de Broadsea.

# 4. Conexión local con BD externa

---

Aunque la base de datos de Eunomia permite satisfacer las necesidades "demo" de la herramienta, en la mayoría de las ocasiones el objetivo final de la implementación de ATLAS es realizar un estudio o investigación específica sobre una base de datos externa, propiedad del usuario o la organización que incopora la herramienta de análisis. Por tanto, es de gran interés presentar cómo establecer una conexión entre ATLAS Broadsea y una base de datos personalizada o propia.

Esta conexión se realiza a través de la WebAPI de Broadsea (véase [1.4](#)) y, en la práctica, se puede realizar de dos formas distintas:

1. La configuración manual, por convención, que utiliza directamente el administrador de la base de datos [4.1](#).
2. La configuración avanzada, por convención, que utiliza los protocolos de seguridad de ATLAS y la webAPI [4.2](#).

El manual presenta ambas opciones aunque solo se seguirá en profundidad la configuración manual, ya que es la opción más sencilla y que los propios desarrolladores recomiendan para una implementación rápida [\[17\]](#). Por otra parte, por la naturaleza del entorno de instalación al que se enfoca este manual (véase [1.1](#)), no es de interés activar el protocolo de seguridad de ATLAS , que debería ser la opción preferente para el despliegue de la herramienta en organizaciones donde haya una necesaria diferenciación entre usuarios administradores con privilegios y usuarios sin privilegios.

## 4.1. Configuración manual

La configuración manual consiste en interactuar con la WebAPI directamente a través de pgAdmin para establecer la conexión con la base de datos externa.

Esta implementación se describe de forma muy completa en la documentación correspondiente al repositorio de github de CDM Configuration [\[7\]](#) y en los foros [\[17\]\[18\]](#).

### 4.1.1. Requisitos previos

El único requisito que se necesita para poder integrar la base de datos externa con la WebAPI de ATLAS es que la base de datos externa esté estructurada según el CDM.

Esto implica una serie de características que se detallan en mayor profundidad en [1.4](#), aunque se presentan otra vez a modo de resumen en este apartado:

1. La base de datos debe estar conformada por los esquemas requeridos por el CDM [7], al menos: cdm y results
2. Los datos y tablas de cdm deben estar normalizados a OMOP.
3. No se requiere que haya un vocabulario explícitamente configurado para la base de datos externa pero debe haber un vocabulario configurado en Broadsea.
4. Por último, no se requiere un estilo de base de datos concreto para la base de datos externa, puede ser Postgre o cualquier otro.

Para realizar esta implementación, según el fin educacional del manual, se utiliza la base de datos de synthea, que reúne todos estos requisitos. Para mayor información sobre la configuración de synthea consultar [4.2.2](#).

#### 4.1.2. Deployment

La integración de la base de datos con la WebAPI es bastante sencilla, basta con configurar los parámetros correspondientes en el esquema webapi de la base de datos de Broadsea. Para ello se utiliza un listado de queries sql extraído de [7]. También se puede descargar el archivo sql completo del repositorio del TFG [1].

1. En primer lugar, se debe añadir la nueva fuente en la tabla sources del esquema webapi. De este modo se registra la nueva fuente y el string jdbc que apunta hacia ella permitiendo la conexión. Para ello se puede utilizar la siguiente query:

---

```

1 INSERT INTO webapi.source (source_id, source_name, source_key,
    source_connection, source_dialect)
2 SELECT nextval('webapi.source_sequence'), 'My Cdm', 'MY_CDM', ' jdbc
    :postgresql://server:5432/cdm?user={user}&password={password}', ' postgresql';

```

---

Donde:

- **source\_id**: Es el identificador de la fuente. Se asigna por defecto.
- **source\_name**: Es el nombre personalizado de la fuente.
- **source\_key**: Es la clave que utilizarán otras tablas para referirse a la fuente.
- **source\_connection**: Es el string de conexión jdbc, que debe seguir la estructura por defecto que se sugiere en la query. Si la
- **source\_dialect**: El dialecto que utiliza la base de datos. Por defecto sería postgresql.

Para el ejemplo de synthea, se aplica la siguiente query:

---

```

1 INSERT INTO webapi.source (source_id, source_name, source_key,
    source_connection, source_dialect)

```

---

---

```

2 SELECT nextval('webapi.source_sequence'), 'synthea1k', 'SYNTHEA1K',
  'jdbc:postgresql://broadsea-atlasdb:5432/synthea1k?user=postgres
&password=mypass', 'postgresql';

```

---

Esto obtiene como resultado global:



The screenshot shows a PostgreSQL query tool interface. The top bar has tabs for 'Query' (selected), 'Query History', and 'Scratch Pad'. Below the tabs is a code editor containing the following SQL query:

```

1 SELECT * FROM webapi.source
2 ORDER BY source_id ASC

```

Below the code editor is a 'Data Output' tab, which is selected. It displays a table with the following data:

	source_id [PK] integer	source_name character varying (255)	source_key character varying (50)	source_connection character varying (8000)	source_dialect character varyin
1	1	OHDSI Eunomia Demo Database	EUNOMIA	jdbc:postgresql://broadsea-atlasdb:5432/postgres?user=postgres&password=mypass	postgresql
2	2	synthea1k	SYNTHEA1K	jdbc:postgresql://broadsea-atlasdb:5432/synthea1k?user=postgres&password=mypass	postgresql

**Figura 4.1:** Captura de pantalla de las fuentes registradas en la tabla sources.

2. En segundo lugar, y por último, se debe registrar cada esquema que utiliza la nueva fuente en la tabla source\_daimon del esquema webapi. Para ello se pueden utilizar las siguientes queries genéricas:
- 

```

1 INSERT INTO webapi.source_daimon (source_daimon_id, source_id,
  daimon_type, table_qualifier, priority)
2 SELECT nextval('webapi.source_daimon_sequence'), source_id, 0, 'cdm'
  ,
  0
3 FROM webapi.source
4 WHERE source_key = 'MY_CDM';
5
6 INSERT INTO webapi.source_daimon (source_daimon_id, source_id,
  daimon_type, table_qualifier, priority)
7 SELECT nextval('webapi.source_daimon_sequence'), source_id, 1, '
  vocab', 1
8 FROM webapi.source
9 WHERE source_key = 'MY_CDM';
10
11 INSERT INTO webapi.source_daimon (source_daimon_id, source_id,
  daimon_type, table_qualifier, priority)
12 SELECT nextval('webapi.source_daimon_sequence'), source_id, 2, '
  results', 1
13 FROM webapi.source
14 WHERE source_key = 'MY_CDM';
15
16 INSERT INTO webapi.source_daimon (source_daimon_id, source_id,
  daimon_type, table_qualifier, priority)
17 SELECT nextval('webapi.source_daimon_sequence'), source_id, 5, 'temp
  ', 0
18 FROM webapi.source
19 WHERE source_key = 'MY_CDM';

```

---

Donde:

- source\_daimon\_id: Es el identificador del daimon. Se asigna por defecto.
- source\_id: Es el identificador de la nueva fuente, según el valor obtenido al registrar la fuente en la tabla sources.
- daimon\_type: Es el identificador que designa tipo de daimon que se está registrando, es decir, el tipo de esquema. Este identificador puede ser "0" para el esquema del cdm, "1" para el vocab, "2" para results y, opcionalmente, "5" para temp.
- table\_qualifier: Es el nombre concreto del esquema al que se está apuntando.
- priority: Es un valor entero mayor o menor que uno que sirve para distinguir cuál es la fuente prioritaria. Para el vocabulario y results, al menos debe haber un daimon con priority  $\geq 1$ .

Para la realización de esta implementación de ejemplo se ha seguido el ejemplo de synthea (4.1.1), que aplica las siguientes queries:

---

```
1 INSERT INTO webapi.source_daimon (source_daimon_id, source_id,
2   daimon_type, tableQualifier, priority)
3 SELECT nextval('webapi.source_daimon_sequence'), source_id, 0, '
4   synthea', 0
5 FROM webapi.source
6 WHERE source_key = 'SYNTHEA1K';
7
8 INSERT INTO webapi.source_daimon (source_daimon_id, source_id,
9   daimon_type, tableQualifier, priority)
10 SELECT nextval('webapi.source_daimon_sequence'), 2, 1, 'cdm', 1
11 FROM webapi.source
12 WHERE source_key = 'SYNTHEA1K';
13
14 INSERT INTO webapi.source_daimon (source_daimon_id, source_id,
15   daimon_type, tableQualifier, priority)
16 SELECT nextval('webapi.source_daimon_sequence'), 2, 2, 'cdm_results'
17   , 0
18 FROM webapi.source
19 WHERE source_key = 'SYNTHEA1K';
```

---

Esto se obtiene como resultado global:

The screenshot shows the ATLAS interface with a query results table. The table has columns: source\_daimon\_id [PK] integer, source\_id integer, daimon\_type integer, table\_qualifier character varying (255), and priority integer. The data shows six rows of results:

source_daimon_id	source_id	daimon_type	table_qualifier	priority
1	1	0	demo_cdm	0
2	2	1	demo_cdm	10
3	3	2	demo_cdm_results	0
4	4	0	cdm	1
5	5	1	demo_cdm	1
6	6	2	cdm_results	0

Figura 4.2: Captura de pantalla de los daimon registrados en la tabla source\_daimon.

Observe que si se agrupase por source\_id hay dos fuentes distintas: (1) la fuente demo que viene preconfigurada con Broadsea y (2) la fuente que se acaba de registrar.

Así de sencillo, ya estaría registrada la nueva fuente, y establecida la conexión con la misma a través de la webAPI de Broadsea.

#### 4.1.3. Comprobación de conexión correcta

Para comprobar que la conexión con la BD externa se ha realizado correctamente y se ha integrado correctamente con la WebAPI y con ATLAS se puede realizar dos estrategias:

1. En primer lugar, acceder a la dirección `http://server/WebAPI/source/refresh` en el buscador del navegador y comprobar que aparece como primera fuente la base de datos de Eunomia y los esquemas que utiliza y, como segunda fuente, la base de datos añadida. El resultado debe ser similar al siguiente código:

---

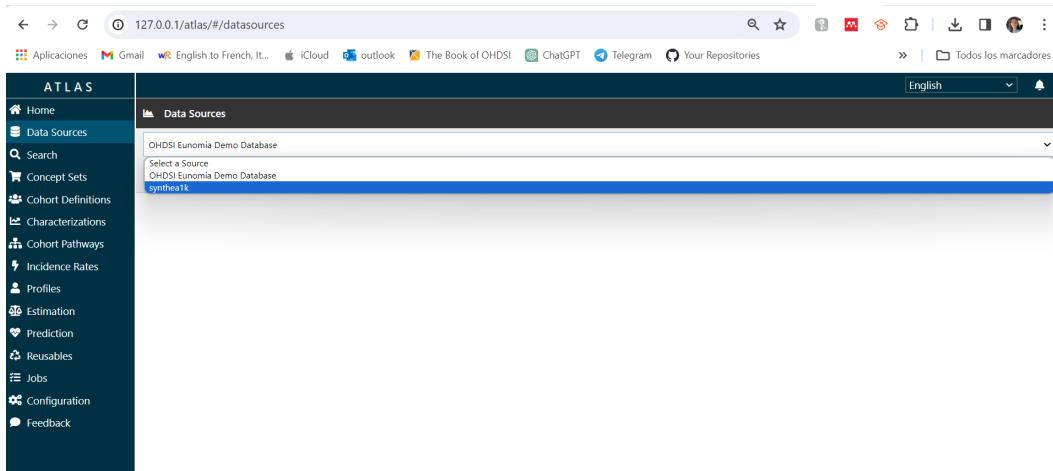
```

1 [{"sourceId":1,"sourceName":"OHDSI Eunomia Demo Database",
  "sourceDialect":"postgresql","sourceKey":"EUNOMIA","daimons":[{
    "sourceDaimonId":1,"daimonType":"CDM","tableQualifier":"demo_cdm",
    "priority":0},{ "sourceDaimonId":3,"daimonType":"Results",
    "tableQualifier":"demo_cdm_results","priority":0},{ "sourceDaimonId":
    2,"daimonType":"Vocabulary","tableQualifier":"omop_vocab",
    "priority":10}]}],
2
3 [{"sourceId":2,"sourceName":"synthea1k","sourceDialect":"postgresql",
  "sourceKey":"SYNTHEA1K","daimons":[{
    "sourceDaimonId":4,"daimonType":"CDM","tableQualifier":"cdm\n",
    "priority":1},{ "sourceDaimonId":5,"daimonType":"Vocabulary",
    "tableQualifier":"omop_vocab","priority":1},{ "sourceDaimonId":6,
    "daimonType":"Results","tableQualifier":"cdm_results",
    "priority":0}]}]

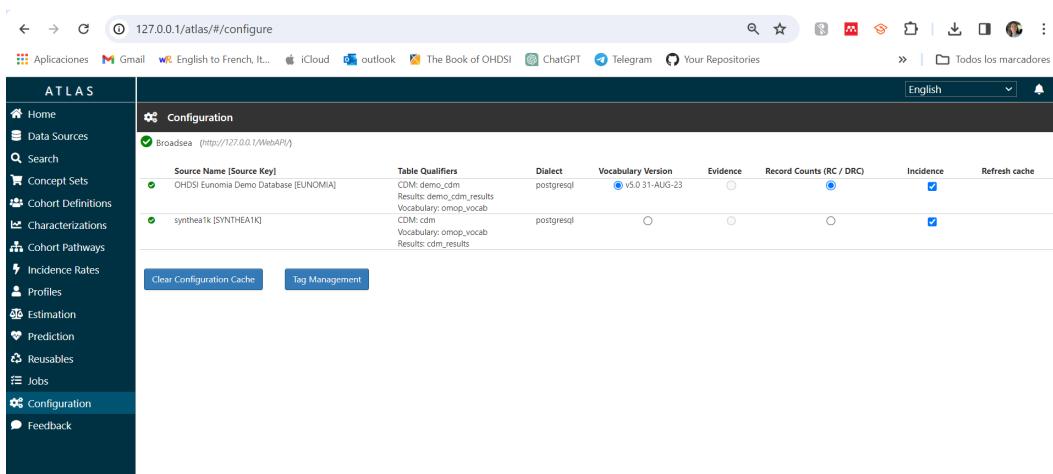
```

---

2. En segundo lugar, comprobar a través de la interfaz gráfica de ATLAS que aparece la nueva fuente, a través del menú desplegable del apartado Data sources o en el apartado Configuration de la herramienta.



**Figura 4.3:** Captura de pantalla del menú Data sources de ATLAS Broadsea



**Figura 4.4:** Captura de pantalla del menú Configuration de ATLAS Broadsea

#### 4.1.4. Solución de posibles problemas

- Faltan esquemas no aparece - Valores mal introducidos

## 4.2. Otras configuraciones avanzadas

### 4.2.1. Configurar la seguridad de Broadsea

Otra forma de establecer conexiones con bases de datos externas es activando el protocolo de seguridad de ATLAS y de la WebAPI. Esta configuración posibilita la adición de fuentes a través de la interfaz gráfica de ATLAS, pues crea en el menú Configuration un nuevo botón para añadir fuentes intuitivamente.



**Figura 4.5:** Imagen extraída de [17] donde aparece el botón para añadir fuentes desde la interfaz gráfica de ATLAS

No obstante, esta forma tan aparentemente sencilla de conexión, verdaderamente se establece a través de protocolos complejos y configuraciones avanzadas de ATLAS y la WebAPI que resultan en grandes cantidades de problemas y errores. De hecho, en varias ocasiones los propios desarrolladores a través de los foros recomiendan su uso solo cuando sea estrictamente necesario [17][19].

Si bien la configuración es compleja, aporta, además de la facilidad de adición de fuentes mediante el botón en ATLAS, la posibilidad de distinguir entre distintos usuarios y roles con distintos permisos y privilegios, lo cual puede ser de gran interés en ciertos entornos u organizaciones. La configuración de la seguridad de ATLAS empieza por la configuración de la seguridad de la WebAPI y ambas están descritas detalladamente en los correspondientes repositorios de Github [20][21]. También hay un foro de gran interés en el que se describen aspectos de utilidad sobre la configuración de la seguridad en una organización [19].

#### 4.2.2. Descargar otras BD

Esta alternativa no es una configuración avanzada en sí misma pero si es una configuración opcional por lo que se encuadra en este marco.

En principio, puede no ser de interés descargar otra base de datos demo si el objetivo del usuario es analizar con ATLAS su propia base de datos, pero puede ser de interés si tan solo se quiere probar el despliegue de la herramienta o su funcionalidad con una base de datos algo más extensa que la la base de datos de Eunomia, que es muy pequeña.

En este caso, precisamente para satisfacer esa necesidad, se ha descargado la base de datos de synthea.

#### Synthea

Realmente, synthea es un generador de datos sintéticos, no una base de datos por sí misma. Synthea permite construir una base de datos con parámetros configurables con información que no es real, pero podría serlo. La documentación de synthea se encuentra en su repositorio de github [22] y en github pages [23]. Los datos de synthea no se crean por defecto estandarizados a OMOP pero OHDSI recoge también en un repositorio de github el rscript necesario para realizar el mapeo [24].

Para la realización de la configuración de la base de datos externa se ha creado manualmente una base de datos en Postgre, dentro del mismo servidor de Broadsea, con los esquemas public, cdm y results.

# 5. Configuración del Vocabulario

---

El vocabulario no viene enteramente configurado en ATLAS Broadsea sino tan solo una pequeña demo adjunta a la base de datos de Eunomia (véase [1.4](#)).

Por tanto, para utilizar ATLAS se requiere la instalación y configuración del Vocabulario más extenso. OHDSI propone, en el repositorio de github de Broadsea [5](#) dos formas de configurar el vocabulario de Broadsea: (1) de forma manual (2) a través de Apache Solr.

## 5.1. Configuración manual

La configuración manual del vocabulario requiere descargar el vocabulario directamente desde ATHENA [\[25\]](#), instalarlo manualmente en el directorio local de Broadsea y ejecutar el contenedor docker encargado de montar el vocabulario en Broadsea.

### 5.1.1. Requisitos previos

Para configurar el vocabulario manualmente, se debe haber descargado el vocabulario y configurado el directorio local de Broadsea. Este procedimiento se ha seguido gracias a los foros "Downloading OMOP cdm version 5 vocabulary data" [\[26\]](#) y "March to the Broadsea" [\[27\]](#) y el tutorial de youtube "Demo: Getting Vocabularies Into My OMOP CDM" [\[28\]](#)

1. Descargar el vocabulario desde la herramienta online de ATHENA. Para ello, acceder al menú de descarga Download. En este momento se presenta un listado de todos los vocabularios que contiene ATHENA entre los que algunos aparecen preseleccionados. Cada usuario puede seleccionar o deseleccionar los vocabularios que le interesen para el estudio que esté realizando. En este caso, se descargará el vocabulario que ATHENA sugiere por defecto.

The screenshot shows a web-based application interface for managing vocabularies. At the top, there's a header bar with various icons and links. Below it is a search bar and a navigation menu. The main content area is titled 'ATHENA' and displays a table of vocabularies. The columns in the table are: ID (CDM V4.5), CODE (CDM V5), NAME, REQUIRED, and LATEST UPDATE. The table lists nine vocabularies, each with a checkbox next to it. The latest update column shows dates ranging from Jan 2022 to Aug 2023. At the bottom of the table, there's a link labeled 'Lirenes'.

ID (CDM V4.5)	CODE (CDM V5)	NAME	REQUIRED	LATEST UPDATE
1	SNOMED	Systematic Nomenclature of Medicine - Clinical Terms (IHTSDO)		28-Jan-2022
2	ICD9CM	International Classification of Diseases, Ninth Revision, Clinical Modification, Volume 1 and 2 (NCHS)		01-Oct-2014
3	ICD9Proc	International Classification of Diseases, Ninth Revision, Clinical Modification, Volume 3 (NCHS)		01-Oct-2014
4	CPT4	Current Procedural Terminology version 4 (AMA)	EULA required	01-May-2023
5	HGPCS	Healthcare Common Procedure Coding System (CMS)		01-Jul-2023
6	LOINC	Logical Observation Identifiers Names and Codes (Regenstrief Institute)		15-Aug-2023
7	NDFRT	National Drug File - Reference Terminology (VA)		06-Aug-2018
8	RxNorm	RxNorm (NLM)		03-Jul-2023
9	NDC	National Drug Code (FDA and manufacturers)		27-Aug-2023

**Figura 5.1:** Captura de pantalla de la preselección de vocabularios para descargar.

2. La descarga requiere registrar un usuario con un correo electrónico válido al que se enviará un link personal que permitirá la descarga de un archivo .zip con el vocabulario seleccionado. También desde ATHENA en la pestaña SHOW HISTORY muestra el estado en el que se encuentra la descarga del vocabulario y, una vez que esté listo, permite la descarga directa del zip.
3. El archivo zip que se descarga, una vez descomprimido, muestra varios archivos .csv con las tablas que conforman el Vocabulario y otros archivos CPT4 que requieren de una configuración adicional. En este caso, no se utilizará el vocabulario de CPT4 por lo que no se realizará esta configuración, sin embargo, viene bien explicada en el tutorial de youtube adjunto al principio de la sección.

```
C:\Users\valle\Downloads\vocabulary_download_v5_{d6368ff7-27c4-41c9-afaf-c0f0764ff716}_1708516913493>dir
El volumen de la unidad C es OS
El número de serie del volumen es: 6416-7C02

Directorio de C:\Users\valle\Downloads\vocabulary_download_v5_{d6368ff7-27c4-41c9-afaf-c0f0764ff716}_1708516913493

21/02/2024 13:43    <DIR>          .
04/03/2024 12:55    <DIR>          ..
21/02/2024 13:41      793.956.864 CONCEPT.csv
21/02/2024 13:42      1.573.127.294 CONCEPT_ANCESTOR.csv
21/02/2024 13:41      17.697 CONCEPT_CLASS.csv
21/02/2024 13:41      1.017.865 CONCEPT_CPT4.csv
21/02/2024 13:43      2.361.915.901 CONCEPT_RELATIONSHIP.csv
21/02/2024 13:41      181.476.737 CONCEPT_SYNONYM.csv
21/02/2024 13:43          48 cpt.bat
21/02/2024 13:43          50 cpt.sh
21/02/2024 13:43          8.737.387 cpt4.jar
21/02/2024 13:41          1.745 DOMAIN.csv
21/02/2024 13:41      158.691.287 DRUG_STRENGTH.csv
21/02/2024 13:43          510 readme.txt
21/02/2024 13:41          51.698 RELATIONSHIP.csv
21/02/2024 13:41          6.138 VOCABULARY.csv
               14 archivos   5.079.001.221 bytes
                  2 dirs     327.704.576 bytes libres
```

**Figura 5.2:** Captura de pantalla de archivos descargados del vocabulario.

4. Por último, los archivos descargados del vocabulario deben almacenarse en el directorio local de Broadsea, concretamente en la ruta Broadsea/omop\_vocab/files. En caso de no existir la carpeta /files, crearla manualmente. Este paso es muy importante.

### 5.1.2. Deployment

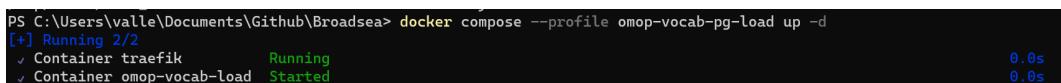
Configurar el vocabulario requiere establecer una configuración avanzada del contenedor Docker de Broadsea. Si bien, la primera vez que se inicializó el contenedor se ejecutó el perfil default ahora se va a ejecutar específicamente el perfil omop-vocab-pg-load. Esta opción se especifica en la sección OMOP Vocab Loading del repositorio de Broadsea. También ha sido de utilidad el repositorio [7] y el mismo foro [27].

Para acceder a la información y configuración avanzada del perfil se puede acceder al docker-compose.yml y a la sección 9 del archivo .env, aunque este caso no será necesario realizar ninguna modificación sobre los mismos.

1. Para comenzar la configuración del vocabulario es necesario inicializar el contenedor omop-vocab-load. Ejecutar la siguiente línea de código en el cmd:

```
1 docker compose --profile omop-vocab-pg-load up -d
```

Este comando da lugar el siguiente resultado.



```
PS C:\Users\valle\Documents\Github\Broadsea> docker compose --profile omop-vocab-pg-load up -d
[+] Running 2/2
  ✓ Container traefik           Running          0.0s
  ✓ Container omop-vocab-load   Started          0.0s
```

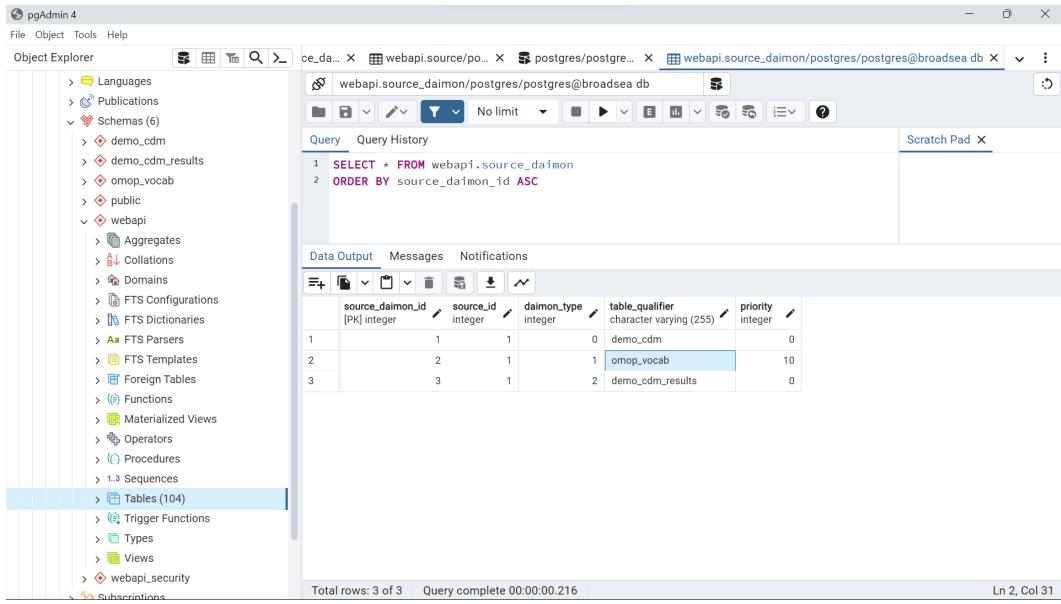
**Figura 5.3:** Captura de pantalla de comando para iniciar el perfil docker.

2. A partir de este momento comienza la instalación del perfil, por lo que es muy importante tener abierto el logs de Docker para poder observar que el proceso se realiza correctamente. El proceso puede durar unas horas y una vez que se finaliza, el log muestra un mensaje que advierte que el contenedor puede ser eliminado y el estado del contenedor pasa a ser "Exited". El resultado es la creación de un nuevo esquema en la base de datos llamado "omop\_vocab" con todos los archivos descargados del vocabulario.
3. El último paso de la configuración es integrar este nuevo esquema a la WebAPI de Broadsea. Este último paso se puede realizar fácilmente de forma manual a través de pgAdmin, abriendo la tabla source\_daimon del esquema webapi de la bd de Broadsea.

La instalación por defecto de Broadsea adjudica la fuente del vocabulario (daimon\_type=1) al demo\_cdm, sin embargo, se debe modificar para que apunte al nuevo esquema omop\_vocab que se ha instalado. Esta modificación se puede realizar directamente haciendo click ssobre la tabla que devuelve la siguiente consulta.

```
1 SELECT * FROM webapi.source_daimon
2 ORDER BY source_daimon_id ASC
```

El resultado final debe ser similar a la siguiente figura.



**Figura 5.4:** Captura de pantalla de la integración final del vocabulario descargado con la webapi

### 5.1.3. Comprobación de configuración correcta

Para comprobar que se ha configurado correctamente el vocabulario en la base de datos de Broadsea, se puede consultar el administrador de la base de datos o realizar directamente una búsqueda en ATLAS.

1. En primer lugar, desde el administrador de la base de datos, pgAdmin, se debe comprobar que el vocabulario se ha instalado correctamente y no se han instalado tablas vacías. Para ello basta con seleccionar cualquier tabla del esquema `omop_vocab` y mostrar todas las filas. Las tablas más significativas son la tabla `vocabulary`, que muestra todos los vocabularios que se han insertado, y la tabla `concept` que reúne todos los conceptos definidos en cada vocabulario. Para esta implementación la primera tabla muestra un total de 59 filas y la segunda, 5.975.392 filas.
2. En segundo lugar, se puede comprobar que la integración del nuevo esquema con el vocabulario haya sido exitosa realizando una búsqueda en el vocabulario directamente desde ATLAS Broadsea.

El vocabulario `demo` que venía incluido con Eunomia presentaba muy pocos conceptos por lo que muchas búsquedas de conceptos no devolvían resultado o devolvían muy pocos. Para este manual, se comprobó que antes de realizar la configuración del vocabulario, al buscar "diabetes" solo se obtenía un resultado, como se muestra en la figura 5.5. Sin embargo, después de configurar correctamente, se obtienen hasta 20602 resultados, como muestra la figura 5.6.

The screenshot shows the ATLAS search interface. The left sidebar contains navigation links such as Home, Data Sources, Search, Concept Sets, Cohort Definitions, Characterizations, Cohort Pathways, Incidence Rates, Profiles, Estimation, Prediction, Reusables, Jobs, Configuration, and Feedback. The Apache 2.0 license information is also present. The main search bar at the top has the query 'DIABETES'. Below it, a table displays one entry: '4144583 427089005 Diabetes mellitus due to cystic fibrosis'. The table includes columns for Id, Code, Name, Class, RC, DRC, PC, DPC, Domain, and Vocabulary. The 'Vocabulary' column shows 'SNOMED (1)'. The 'Name' column shows 'Diabetes mellitus due to cystic fibrosis'. The 'Class' column shows 'Clinical Finding'. The 'Domain' column shows 'timeout'. The 'PC' column shows 'timeout'. The 'DPC' column shows 'timeout'. The 'Domain' column shows 'Condition'. The 'Vocabulary' column shows 'SNOMED'. At the bottom, there are buttons for Exclude, Descendants, Mapped, Preview, and Add to New Concept Set.

**Figura 5.5:** Captura de pantalla de la búsqueda en el vocabulario antes de la configuración

The screenshot shows the ATLAS search interface after configuration. The search bar now contains 'diabetes'. The table below shows 50 entries out of 2,602. The first few results are: '403872 190484000 (Diabetes insipidus [& pituitary]) or (vasopressin deficiency)', '(Renal glycosuria) or (renal diabetes)', '41504196 OMOP2702158 80000 MG Calendula officinalis extract 0.02 MG/MG / Centaunum erythraea preparation 0.05 MG/MG / Cynara scolymus leaf extract 0.04 MG/MG / Fucus vesiculosus extract 0.06 MG/MG / - Oral Solution [Diabetette] Box of 1', '41504137 OMOP2702099 80000 MG Calendula officinalis extract 0.02 MG/MG / Centaunum erythraea preparation 0.05 MG/MG / Cynara scolymus leaf extract 0.04 MG/MG / Fucus vesiculosus extract 0.06 MG/MG / Juglans regia leaf extract 0.03 MG/MG / - Oral Solution [Diabetette]', '35609948 1087971000000105 ABCD2 (Agn, Blood pressure, Clinical features, Duration, Diabetes) score', '4034964 237620003 Abnormal metabolic state due to diabetes mellitus', '42536400 735200002 Absence of lower limb due to diabetes mellitus', and '4321756 9859006 Acanthosis nigricans due to type 2-diabetes mellitus'. The table structure is identical to Figure 5.5, with columns for Id, Code, Name, Class, RC, DRC, PC, DPC, Domain, and Vocabulary.

**Figura 5.6:** Captura de pantalla de la búsqueda en el vocabulario después de la configuración

3. Por último, se puede realizar una comprobación extra accediendo a la dirección <http://127.0.0.1/WebAPI/source/refresh> en el navegador, que imprime en pantalla las fuentes que se están utilizando en la configuración de Broadsea. El resultado debe ser similar al siguiente, prestando especial atención a que aparezca `omop_vocab` como fuente del vocabulario:

```

1  [{"sourceId":1,"sourceName":"OHDSI Eunomia Demo Database","sourceDialect":"postgresql","sourceKey":"EUNOMIA","daimons":[{"sourceDaimonId":1,"daimonType":"CDM","tableQualifier":"demo_cdm","priority":0},{"sourceDaimonId":3,"daimonType":"Results","priority":1}]}

```

```
tableQualifier":"demo_cdm_results","priority":0}, {"sourceDaimonId":2,"daimonType":"Vocabulary","tableQualifier":"omop_vocab","priority":10}]]]
```

---

### 5.1.4. Solución de posibles errores

#### Error: can't cd to /temp/files: No such files or directory

Al inicializar el perfil omop-vocab-load el contenedor no se construye correctamente y el log muestra el siguiente aviso, tal y como se muestra en la figura.

```
PS C:\Users\valle\Documents\Github\Broadsea> docker logs omop-vocab-load
fetch https://dl-cdn.alpinelinux.org/alpine/v3.17/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.17/community/x86_64/APKINDEX.tar.gz
(1/8) Installing postgresql-common (1.1-r3)
Executing postgresql-common-1.1-r3.pre-install
(2/8) Installing tz4-libs (1.9.4-r1)
(3/8) Installing libpq (15.6-r0)
(4/8) Installing ncurses-terminfo-base (6.3_p20221119-r1)
(5/8) Installing ncurses-libs (6.3_p20221119-r1)
(6/8) Installing readline (8.2.0-r0)
(7/8) Installing zstd-libs (1.5.5-r0)
(8/8) Installing postgresql15-client (15.6-r0)
Executing busybox-1.35.0-r29.trigger
Executing postgresql-common-1.1-r3.trigger
* Setting postgresql15 as the default version
OK: 12 MiB in 23 packages
/tmp/scripts/load_omop_vocab.sh: cd: line 7: can't cd to /tmp/files: No such file or directory
```

**Figura 5.7:** Captura de pantalla de la búsqueda en el vocabulario después de la configuración

Solución: Crear manualmente la carpeta /files

#### Error: El contenedor se construye correctamente pero se construye vacío

Al inicializar el perfil omop-vocab-load el contenedor se construye en pocos segundos aparentemente correctamente. Incluso es posible que aparezca en el administrador de la base de datos pero aparece el esquema vacío o las tablas del esquema vacías, sin filas.

Solución: La carpeta /files debe contener los archivos .csv que se han descargado de ATHENA.

## 5.2. Otras configuraciones avanzadas

En este punto el vocabulario ya está perfectamente instalado y configurado, sin embargo, Broadsea ofrece más opciones avanzadas para sacar aún mayor partido de la herramienta. En este manual se presentan dichas opciones aunque no se procederá a su configuración.

### **5.2.1. Configuración de Apache Solr**

ATLAS Broadsea hace uso de Apache Solr a través de diferentes perfiles docker para facilitar la importación del vocabulario y la búsqueda en el mismo. Sin embargo, ambas características aún se encuentran en desarrollo, según la información facilitada en el repositorio de github de Broadsea Solr [29].

El proceso de configuración de Broadsea Solr se encuentra en aquel mismo repositorio y la descripción de los perfiles que ejecutan los contenedores Solr se presentan en el de Broadsea [4].

1. Con el perfil `solr-vocab-with-import` se puede automatizar la importación del vocabulario desde el servidor de Apache en vez de tener que configurarlo manualmente. Sin embargo, esta opción que parece facilitar el trabajo al usuario en la práctica resulta engorrosa, por ello no ha sido cubierta en este manual.
2. El otro perfil, `solr-vocab-no-import` simplemente lanza el servidor Apache "vacío" permitiendo la configuración manual del vocabulario desde la GUI de Apache Solr.

Para mayor información o resolución de conflictos consultar los foros de OHDSI o los "issue" del repositorio de github.

### **5.2.2. Configuración de PHOEBE**

ATLAS Broadsea puede configurarse junto a PHOEBE, que es una herramienta OHDSI, también disponible [online](#), que consiste en un recomendador de conceptos del vocabulario, de forma que facilite la construcción de conjuntos de conceptos durante la realización de estudios con ATLAS.

No existe un repositorio concreto de esta herramienta pero el perfil que implementa el contenedor docker se presenta brevemente en el repositorio de [Broadsea](#). También se puede encontrar información relevante en algunos foros de OHDSI como en el de [Phoebe 2.0](#).

De igual forma, para mayor información o resolución de conflictos consultar otros foros de OHDSI o los "issue" del repositorio de github de Broadsea o ATLAS.

# Bibliografía

---

- [1] vallealonsodc. Thesis-ATLAS-OHDSI. <https://github.com/vallealonsodc/Thesis-ATLAS-OHDSI>, sin fecha.
- [2] Ajit et al Londhi. Broadsea 3.0: “broadening the ohdsea”. 2023. URL [https://www.ohdsi.org/wp-content/uploads/2023/10/Londhe-Ajit\\_Broadsea-3.0-BROADening-the-ohdSEA\\_2023symposium-Ajit-Londhe.pdf](https://www.ohdsi.org/wp-content/uploads/2023/10/Londhe-Ajit_Broadsea-3.0-BROADening-the-ohdSEA_2023symposium-Ajit-Londhe.pdf).
- [3] A. Londhe. Slides from the 2023 ohdsi global symposium, 2023. URL <https://www.ohdsi.org/wp-content/uploads/2023/10/419-Londhe-Slides.pdf>.
- [4] OHDSI. Broadsea, 2023. URL <https://github.com/OHDSI/Broadsea>.
- [5] OHDSI. Ohdsi webapi, 2023. URL <https://github.com/OHDSI/WebAPI>.
- [6] OHDSI. Ohdsi webapi wiki, 2023. URL <https://github.com/OHDSI/WebAPI/wiki>.
- [7] OHDSI. Ohdsi common data model configuration (wiki), 2023. URL <https://github.com/OHDSI/WebAPI/wiki/CDM-Configuration>.
- [8] OHDSI. Ohdsi common data model, 2023. URL <https://github.com/OHDSI/CommonDataModel>.
- [9] OHDSI. Ohdsi common data model (github pages), 2023. URL <https://ohdsi.github.io/CommonDataModel/>.
- [10] OHDSI. *The Book of OHDSI*. 2023. URL <https://ohdsi.github.io/TheBookOfOhdsi/>.
- [11] OHDSI. Ohdsi vocabulary v5.0 (wiki), 2023. URL <https://github.com/OHDSI/Vocabulary-v5.0/wiki>.
- [12] OHDSI. Ohdsi broadsea atlasdb, 2023. URL <https://github.com/OHDSI/Broadsea-Atlasdb>.
- [13] OHDSI. Ohdsi eunomia, 2023. URL <https://github.com/OHDSI/Eunomia>.
- [14] OHDSI. Ohdsi eunomia (github pages), 2023. URL <https://github.com/OHDSI/Eunomia>.
- [15] OHDSI. Ohdsi atlas demo, 2023. URL <https://atlas-demo.ohdsi.org/>. Accessed from the OHDSI Atlas demo platform (dynamic web page).
- [16] OHDSI Forum. Question about broadsea default database, 2023. URL <https://forums.ohdsi.org/t/question-about-broadsea-default-database/18431>.

- [17] OHDSI Forum. Adding a new ms sql cdm database to atlas using broadsea, . URL <https://forums.ohdsi.org/t/adding-a-new-ms-sql-cdm-database-to-atlas-using-broadsea/19404/3>.
- [18] OHDSI Forum. Broadsea 3.0 quick start: How to add my own db in atlas omop, . URL <https://forums.ohdsi.org/t/brodsea-3-0-quick-start-how-to-add-my-own-db-in-atlas-omop/20617/21>.
- [19] OHDSI Forum. Adding security in atlas, . URL <https://forums.ohdsi.org/t/adding-security-in-atlas/6542/13>.
- [20] OHDSI. Webapi security configuration, 2023. URL <https://github.com/OHDSI/WebAPI/wiki/Security-Configuration>.
- [21] OHDSI. Atlas security, 2023. URL <https://github.com/OHDSI/WebAPI/wiki/Atlas-Security>.
- [22] Synthetic Health. Synthea, 2023. URL <https://github.com/synthetichealth/synthea>.
- [23] Synthetic Health. Synthea (github pages), 2023. URL <https://github.com/synthetichealth/synthea>.
- [24] OHDSI. Etl-synthea, 2023. URL <https://github.com/OHDSI/ETL-Synthea>.
- [25] OHDSI. Athena search terms, 2024. URL <https://athena.ohdsi.org/search-terms/start>. Accessed from the OHDSI Athena platform (dynamic web page).
- [26] OHDSI Forums. Downloading omop cdm version 5 vocabulary data. <https://forums.ohdsi.org/t/downloading-omop-cdm-version-5-vocabulary-data/3321/3>, September 18 2017.
- [27] OHDSI Forum. March to broadsea, 2023. URL <https://forums.ohdsi.org/t/march-to-the-broadsea/20576>.
- [28] OHDSI Youtube. Demo: Getting vocabularies into my omop cdm. <https://www.youtube.com/watch?v=FCHxAQOBptE>, November 9 2021.
- [29] OHDSI. Broadsea solr, 2023. URL <https://github.com/OHDSI/Broadsea-Solr>.