

UNIVERSIDADE DE SANTIAGO DE
COMPOSTELA



ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

Seguimento de contaminación de ciudades españolas a partir de datos de TROPOMI

Autor/a:

Hugo Gómez Sabucedo

Titores:

Joaquín Ángel Triñanes Fernández

Grao en Enxeñaría Informática

Febreiro 2024

Traballo de Fin de Grao presentado na Escola Técnica Superior de Enxeñaría
da Universidade de Santiago de Compostela para a obtención do Grao en
Enxeñaría Informática



D. Joaquín Ángel Triñanes Fernández, Profesor/a do Departamento de Electrónica e Computación da Universidade de Santiago de Compostela,

INFORMA:

Que a presente memoria, titulada (*Seguimento de contaminación decidades españolas a partir de datos de TROPOMI*), presentada por **D. Hugo Gómez Sabucedo** para superar os créditos correspondentes ao Traballo de Fin de Grao da titulación de Grao en Enxeñaría Informática, realizouse baixo nosa tutoría no Departamento de Electrónica e Computación da Universidade de Santiago de Compostela.

E para que así conste aos efectos oportunos, expiden o presente informe en Santiago de Compostela, a (Data):

Titor/a,

Alumno/a,

(Joaquín Ángel Triñanes Fernández) (Hugo Gómez Sabucedo)

Agradecimentos

Se se quiere pór algún agradecimento, este vai aquí.

Resumo

Breve resumo das principais contribucións do traballo.

Índice xeral

1. Introducción	1
1.1. Obxectivos do traballo	2
1.2. Estrutura da memoria	2
1.3. Descrición do sistema	3
2. Especificación de Requisitos	5
2.1. Identificación dos usuarios	5
2.2. Requisitos funcionais	6
2.2.1. Obter datos de satélite da ESA [RF1]	6
2.2.2. Transformar e agregar datos de satélite [RF2]	6
2.2.3. Implementar un servidor ERDDAP local [RF3]	6
2.3. Requisitos do produto	7
2.3.1. A aplicación deberá ser rápida e eficiente [RNF1]	7
2.3.2. Eficiencia de almacenamento [RNF2]	7
2.4. Requisitos de rendemento	7
2.4.1. Tempo de resposta razoable [RNF3]	7
3. Deseño	9
3.1. Componentes do sistema	9
3.1.1. Backend	10
3.1.2. Middleware	12
3.1.3. Frontend	12
3.2. Tecnoloxías empregadas	13
3.2.1. Anaconda	13
3.2.2. HARP	13
3.3. Desenvolvemento do sistema	13
3.3.1. Introducción	13
3.4. Implementación do sistema	14
4. Probas	15
5. Conclusións e posibles ampliacións	17

A. Manuais técnicos	19
B. Manuais de usuario	21
Bibliografía	23

Índice de figuras

Índice de cadros

Capítulo 1

Introdución

O cambio climático é un asunto que, actualmente, está á orde do día. O aumento das temperaturas a nivel global é un asunto que preocupa non só ós científicos, senón tamén á poboación en xeral. As causas do cambio climático son moitas, pero podemos estar de acordo en que unha gran parte é debida á presenza de múltiples gases ou substancias químicas que afectan á atmosfera.

Todos estes axentes afectan non só ó medio ambiente, senón ós seres humanos. Por iso, monitorizar a súa evolución é algo crucial, especialmente en entornos especialmente críticos, coma poden ser as cidades, nas cales a conxunción dunha gran densidade de poboación e movementos en áreas, polo xeral, pequenas, fai que sexa crucial coñecer cal é o nivel destes axentes contaminantes. Desta forma, as autoridades poden adoptar medidas, tanto no eido da saúde pública, para previr enfermidades asociadas a unha elevada presenza de ditos contaminantes, como de protección do medio ambiente, para evitar que se causen danos ó entorno no que vivimos.

Así, dende a Axencia Espacial Europea (ESA) lanzouse, no ano 2017, un satélite de observación terrestre nomeado Sentinel-5 Precursor (ou Sentinel-5P). Dito satélite, que se enmarca dentro do Programa Copernico, emprega un instrumento denominado espectrómetro, co obxectivo de monitorizar a contaminación atmosférica. Desta forma, TROPOMI proporciona observacións de forma diaria de gases atmosféricos que son chave á hora de monitorizar a calidade do aire e realizar predicións sobre a evolución da mesma, coma poden ser o metano (CH_4), monóxido de carbono (CO), formaldehído ($HCHO$), dióxido de nitróxeno (NO_2), ozono (O_3), aerosol ou dióxido de xofre (SO_2).

Este satélite proporciona datos a dous niveis distintos: L1, con datos crus; e L2, con variables xeofísicas derivadas dos datos de nivel 1. Existe outro nivel, o Nivel 3 (L3), no cal as variables están mapeadas en escalas regulares. Será este o nivel que nos interese alcanzar nos nosos datos, para poder obter con eles un agregado mensual que nos permita ver a media ou crear climatoloxías para poder observar anomalías nos mesmos.

O obxectivo deste traballo é, por tanto, empregar os datos que nos propor-

ciona este satélite para, mediante a realización de diferentes operacións sobre os mesmos, poder facer un seguimento da evolución ó longo do tempo da contaminación en diversas cidades de España. Decidiuse escoller este ámbito debido a que, ó tratarse dun área relativamente reducida, o procesamento dos datos será máis sinxelo, posto que será preciso procesar menos arquivos de menor tamaño.

1.1. Obxectivos do traballo

Os obxectivos deste traballo de fin de grao son os seguintes:

1. Mapear a contaminación atmosférica a partir de datos de sensores remotos, con especial relevancia en entornos urbanos
2. Monitorizar a variabilidade e tendencia dos diferentes parámetros satelitais relacionados coa contaminación do aire.
3. Desenvolver un visor online que permita representar de forma dinámica a información.
4. Implementar un sistema interoperable de distribución dos datos e produtos.

1.2. Estrutura da memoria

A presente memoria estrutúrase do seguinte xeito:

- **Capítulo 2:** Especificación de requisitos. Este capítulo describe detalladamente os requisitos e criterios que debe cumprir o proxecto.
- **Capítulo 3:** Deseño. Aquí explicaremos todo o relativo ó proceso de deseño do software ata acadar a súa versión final. Tamén se comenta a metodoloxía empregada para o seu desenvolvemento, así como as diferentes tecnoloxías das que se fixo uso no mesmo.
- **Capítulo 4:** Probas. Esta sección trata sobre as probas realizadas, así como os resultados das mesmas.
- **Capítulo 5:** Conclusións e posibles ampliacións. Neste último capítulo preséntanse as conclusións obtidas após realizar o proxecto, así como propostas de melloras sobre o mesmo ou posibles ampliacións futuras.

1.3. Descrición do sistema

Como se explicou nos obxectivos do traballo, o sistema que deseñemos terá como finalidade poder monitorizar a variabilidade e tendencia de diferentes parámetros, obtidos a través de datos de satélite, os cales se empregan para medir a contaminación do aire.

Será preciso, por tanto, deseñar un módulo mediante o cal poidamos descargar os datos de TROPOMI, obtidos a través da ESA, para poder traballar con eles, realizando as transformacións anteriormente mencionadas. Así, empregando as APIs proporcionadas, poderemos deseñar unha función que busque un determinado produto (por exemplo, NO_2) no catálogo de COPERNICUS, filtrando os resultados para centrarnos nun área de interese (neste caso, a Península Ibérica) e datas concretas. Construindo unha query, e empregando a librería `requests`, descargaremos os arquivos para, posteriormente, transformalos. Ademais, os arquivos veñen en formato `.zip`, polo que será preciso, neste módulo, descomprimilos para poder obter os arquivos de formato `.nc` cos que debemos traballar. Este módulo será desenvolvido na linguaxe Python.

Por outra parte, deseñaremos outro módulo, tamén en Python, no cal poidamos traballar cos arquivos descargados. Os datos que descargamos son de nivel 2, polo que temos que realizar operacións cos mesmos para transformalos en datos agregados por mes. Para isto, será preciso empregar `HARP`, unha ferramenta que permite ler e procesar datos de satélite, permitindo a inxesta de arquivos e o traballo cos mesmos. Así, importaremos tódolos arquivos descargados e realizaremos as transformacións anteriormente citadas, para posteriormente exportar o resultado nun arquivo, tamén de formato `.nc`, nun directorio específico.

Os datos transformados almacenaranse nun servidor ERDDAP, que empregaremos como middleware. Este servidor proporciona unha forma sinxela de acceder a datasets científicos en formatos de arquivos comúns, facilitando a xeración de gráficos e mapas. Unifica diferentes tipos de servidores con diferentes estruturas e devolve os datos nun formato de arquivo común, así como estandariza as datas entre os mesmos. Ademais, é posible configurar un servidor ERDDAP propio e empregar os nosos propios datos. Esta será a opción que elixamos, xa que nos permite empregar os datos que nós mesmos transformamos.

Será preciso establecer, no servidor onde corra a aplicación, unha tarefa automática, para que, segundo se publiquen datos novos, estes estean dispoñibles na nosa aplicación. Esta tarefa deberá executarse tódolos meses, tendo en conta que os datos das medicións de satélite tardan uns días en estaren dispoñibles, e deberá descargar os datos dos parámetros que empregue a nosa aplicación, transformalos e almacenalos no directorio establecido.

Por último, deseñaremos unha interface web que permitirá consultar a evolución de cada un dos diferentes parámetros (escollendo un deles) ó longo do tempo.

Todo isto implementarémoslo nunha máquina de AWS, de forma que o sistema

se administrará dende a mesma. Así, poderemos acceder á interface web de forma sinxela, a través de internet, e toda a xestión da aplicación estará tamén centralizada. Nesta máquina almacenaremos os datos transformados que obteñamos do módulo de Python, e nela teremos tamén o servidor ERDDAP co cal acceder a ditos datos. Accedendo á máquina poderemos visualizar a interface web que mencionamos.

Capítulo 2

Especificación de Requisitos

Neste capítulo trataremos en profundidade os principais requisitos que debe cumprir o sistema e o traballo; isto é, as funcionalidades que se require que teña o sistema. Especificaremos os usuarios que empregarán o sistema, as funcionalidades que se lle atribúen a cada un e os compoñentes do sistema, así como os requisitos funcionais e non funcionais do mesmo.

2.1. Identificación dos usuarios

O obxectivo deste proxecto é proporcionar información de diferentes parámetros relacionados coa contaminación atmosférica, co obxectivo de poder realizar un seguimento de como evolucionou ó longo do tempo a contaminación en diferentes puntos da Península Ibérica. Temos, por tanto, un perfil de usuario claro: aquel que consultará os datos, ou usuario final [U1]. Por outra parte, identificaremos outro usuario, administrador [U2] que será o encargado de realizar o mantemento de todo o sistema, e que se encargará de arranxar os distintos problemas que poidan surxir co uso do mesmo.

- O **usuario final** ([U1]) é aquela persona á que vai dirixida principalmente o sistema, e que ten interese en poder monitorizar a evolución da contaminación por diferentes parámetros ó longo do tempo. Trátase dun perfil de persoa variada, dende un científico, un traballador público que queira elaborar plans para a redución da contaminación nas cidades, ou calquera cidadán con interese polo tema. Porén, as tarefas que se lle atribúen son sinxelas:
 1. **Consultar o nivel de contaminación.** Este usuario consultará, na interface web do servidor en que se desenvolva o sistema, o valor dun parámetro en concreto, de entre tódolos que estean dispoñibles, para un mes concreto.

- **O administrador do sistema [U2]** é a persoa que se encargará do mantemento do mesmo, monitorizando o funcionamento do servidor para que, en caso de que se produza algún erro, poder solventalo. O rol de administrador do sistema asumirá o alumno encargado de desenvolver este traballo. As súas principais tarefas serán:
 1. **Configurar o sistema.** Este usuario será o encargado de realizar a configuración inicial do sistema, implementando os diferentes módulos que o compoñen e descargando e procesando datos durante un periodo de tempo o suficientemente longo, que permita a correcta utilización do sistema.
 2. **Monitorear o sistema.** Como administrador, deberá encargarse de monitorizar o correcto funcionamento do sistema, vixiando os posibles problemas que poidan xurdir no mesmo. Especialmente, deberá facer énfase en controlar que, cada mes, se descarguen e procesen correctamente os arquivos correspondentes, para asegurar a máxima calidade nos datos que consulte o usuario final.

2.2. Requisitos funcionais

2.2.1. Obter datos de satélite da ESA [RF1]

A aplicación que se desenvolva debe ser capaz de obter, de forma automática, os datos de satélite dispoñibles no dataspace de COPERNICUS, na web da Axencia Espacial Europea. Isto farase empregando unha API de entre aquelas que ofrece a propia ESA. Estes datos deberán limitarse a unha área xeográfica de interese, que no noso caso será a Península Ibérica, e a un rango de datas determinados.

2.2.2. Transformar e agregar datos de satélite [RF2]

Os datos que se descarguen dende a web da ESA deberán ser transformados, de forma que serán agregados mensualmente para poder facer un seguemento da evolución da contaminación. Os arquivos resultantes deberán manter unicamente a información necesaria.

2.2.3. Implementar un servidor ERDDAP local [RF3]

No servidor onde se implemente este sistema, implementárase un servidor ERDDAP local, de forma que se facilite o acceso ós datos. Deberase configurar un *dataset* por cada un dos parámetros, e almacenar nel os diferentes arquivos.

2.3. Requisitos do produto

2.3.1. A aplicación deberá ser rápida e eficiente [RNF1]

A aplicación deberá ser capaz de, nun tempo razoable dende que estean dispoñibles, descargar e procesar os datos de satélite, de forma que os usuarios poidan acceder a estos datos o antes posible.

2.3.2. Eficiencia de almacenamento [RNF2]

A aplicación debe ser eficiente en canto ó uso do almacenamento se refire. Tendo en conta que, aproximadamente, cada mes se descargarán en torno a 50GB de novos datos por cada produto que se procese, e que após o procesamento teremos datos cun tamaño resultante darredor de 500MB por parámetro, é crucial que a aplicación almacene só os datos estritamente necesarios.

2.4. Requisitos de rendemento

2.4.1. Tempo de resposta razoable [RNF3]

A aplicación debe ter un tempo de resposta razoable, de forma que os usuarios non deban agardar moito tempo para teren os resultados que desexan.

Capítulo 3

Deseño

Neste capítulo, explicaremos os diferentes compoñentes que forman o sistema e as tecnoloxías empregadas, así como o proceso de desenvolvemento e implementación do mesmo.

3.1. Compoñentes do sistema

No que respecta á arquitectura do sistema software desenvolvido, podemos identificar tres grandes compoñentes:

- **Backend.** O backend da aplicación está formado polos diferentes scripts de Python que se encargan da descargar os datos, mediante a API, da web de COPERNICUS, da súa descompresión, o procesamento e o almacenamento no directorio apropiado para que se poida acceder a eles. Ademais, cun script de Python e o programa `cron` de Ubuntu, definiremos unha tarefa a executarse automaticamente de forma mensual, de forma que, segundo teñamos dispoñibles novos datos, estes se descarguen e sexan accesibles ó usuario, para que teña sempre os datos máis recentes.
- **Middleware.** O middleware está formado polo servidor ERDDAP, que serve como unha capa intermedia entre o backend de Python e o frontend que verá o usuario final. Ademais, como comentamos na introdución, proporciona unha forma sinxela de acceder a datasets científicos en formatos de arquivos comúns, facilitando a xeración de gráficos e mapas. Coa implementación do noso propio servidor ERDDAP, poderemos almacenar nel, ademais dos datasets que xa veñen por defecto, uns cos nosos arquivos.
- **Frontend.** O frontend desenvolverase empregando HTML. Será un frontend sinxelo, de forma que, cando se acceda á aplicación na URL do servidor, o usuario vexa un mapa, acoutado á zona na que se realiza o estudo dos datos, e que poida elixir, por unha parte, un parámetro de entre os catro

que se proporcionan, e un mes no rango de meses dispoñibles no dataset, actualizándose a información de forma dinámica.

3.1.1. Backend

Como xa comentamos, o backend da aplicación desenvolveuse en Python. A elección desta linguaxe de programación para este módulo é sinxela: é a linguaxe de programación máis axeitada para traballar con grandes cantidades de datos. Ademais, é a linguaxe que ten maior compatibilidade con tódalas ferramentas que deberemos empregar para o procesado dos datos, o que fai que sexa, practicamente, a única opción a ter en conta..

Así, o backend de Python divídese en catro scripts diferentes, cada un cunha funcionalidade determinada.

Script principal e automatización

Por unha parte, temos o script principal, `app.py`. Este é o script dende o que se executa o programa. Nel, establécense os parámetros cos cales se executaran os módulos de descarga e procesado dos arquivos, comprobando tamén a súa validez (como, por exemplo, que a data de inicio da busca sexa anterior á data de fin). Pódese executar tanto por liña de comandos coma importando o arquivo e chamando á función.

Por outra parte, temos o script `auto.py`, o cal se encarga da automatización da descarga e procesado dos datos. Este será o script que executaremos con `cron` de forma mensual, para manter os datasets actualizados. Este script obtén, a partir da data actual, as datas para as que realizaremos a busca, que serán entre o primeiro e último día do mes anterior. Desta forma, executaremos o `job` de `cron`, cunha frecuencia mensual, o día 10 de cada mes. Polo tanto, no caso da execución que se realizaría o día 10 de xaneiro de 2024, as datas que se tomarían como referencia serían o 1 de decembro de 2023 como data de inicio, e o 31 de decembro de 2023 como data de fin. Decidiuse elixir o décimo día de cada mes para a actualización dos datos debido a que, como se explica en [1], o procesado dos arquivos resultantes das medicións dos satélites ten unha latencia de 5 días. Así, asegurámonos de que, cando se execute o script, tódolos arquivos que se deben ter en conta estarán dispoñibles.

Script de descarga de datos

En segundo lugar, temos o script de descarga de datos. A súa función principal é `obtenArquivos`, que recibe como parámetros a area de interese da busca, as datas de inicio e de fin, o parámetro para o que se realizará a busca e o directorio onde se almacenarán os arquivos. Se este directorio non existe, creao.

A continuación, empregando a API de oData, realiza unha consulta para obter os arquivos que se descargarán. Iterando por estes arquivos, obtén o seu Id e o nome e, se estes non existen (tanto en formato zip, por estaren comprimidos, ou en formato zip, por estaren descomprimidos), realiza a súa descarga, na función `descargaArchivo`. Posteriormente, descomprime o arquivo coa función `descomprimeArchivo`, os cales quedan dentro dunha carpeta co seu nome. Por tanto, unha vez descargados e descomprimidos tódolos arquivos, móveos á carpeta principal e elimina tódalas subcarpetas.

No que respecta á función de descarga de arquivos, en primeiro lugar chama a unha función para obter o token de autorización preciso para descargar os datos. A continuación, crea a url para a descarga e, mediante a librería `Session`, descarga o arquivo, sempre que a resposta do servidor sexa positiva. Por outra parte, a función de descompresión dos arquivos comproba, en primeiro lugar, que o ficheiro zip exista. Se é así, empregando a librería `zipfile`, extrae os arquivos no directorio (imprimindo unha mensaxe de erro en caso de que o arquivo sexa inválido) e, unha vez feito, bórrao.

Script de procesado de datos

Por último, temos o script que se encarga de procesar os datos. Nel, o módulo principal que empregamos é `HARP` que, como se explica en 3.2.2, é unha ferramenta que permite a lectura e procesado de datos de satélite. Neste módulo temos diferentes funcións de procesado, unha para cada un dos diferentes parámetros que se ofrecen na aplicación, xa que as operacións deben personalizarse en función do parámetro. Así, establécese unha función xenérica que, en función do parámetro que se estea tratando, invoca a unha función ou outra e, unha vez procesados tódolos arquivos e exportados os datos, elimina os arquivos que se descargaron, co obxectivo de optimizar o almacenamento do servidor (posto que estamos falando de centenas de arquivos cada mes, con tamaños de entre 500MB e 1GB, os cales, unha vez procesados, non son de interese nin utilidade). Ademais, temos unha función denominada `obtenListaProductos`, para obter os arquivos a procesar.

Malia que cada unha das funcións de procesado son bastante semellantes, posto que realizan unha serie de operacións similares entre elas, foi preciso particularizalas, xa que o nome das variables que exportaremos difire en función do parámetro. Estas funcións reciben unha lista de arquivos a procesar e un directorio onde almacenar o arquivo final procesado. Empregando a función `import_product` de `harp`, importamos os arquivos, realizando con eles unha serie de operacións para filtrar os datos segundo a súa calidade, como se indica nos diferentes manuais. Unha vez temos tódolos produtos importados, realizamos unha serie de operacións, propias para cada produto, para quedarnos cos datos en determinadas lonxitudes e obter con eles unha cuadrícula; e unha serie de post-operacións, que son globais a tódalas funcións, para realizar a agregación dos datos. A continuación, definimos o nome do arquivo a exportar (que segue o patrón parame-

tro_AAAAMM.nc), e gardámolo.

3.1.2. Middleware

O middleware que empregamos será ERDDAP. Isto é un servidor de datos que proporciona unha forma simple e consistente de acceder e descargar múltiples datasets científicos, de diferentes fontes, nun formato común, facilitando a elaboración de gráficos e mapas [2]. Ademais, permite obter os datos en diversos formatos de arquivo, ademais de proporcionar funcións para estandarizar as datas dos resultados ou proporcionar os mesmos en formatos de imaxe personalizados. A instalación de ERDDAP de referencia é aquela feita pola National Oceanic and Atmospheric Administration (NOAA, Oficina Nacional de Administración Oceánica e Atmosférica) que, ademais de ter máis de 200 datasets, permite crear un servidor local para almacenar nel datos propios.

Para realizar esta instalación, só será necesario seguir os pasos que se indican en [3], o que explicaremos máis detalladamente en 3.3 e 3.4. En resumo, é preciso:

1. Dispoñer dunha versión de Java 17.
2. Configurar Tomcat.
3. Descargar os arquivos de configuración de ERDDAP e realizar os cambios precisos para adaptalo ó sistema.
4. Instalar o arquivo `.war` de ERDDAP.
5. Iniciar o servidor e configurar os datasets que queremos que sexan visibles no mesmo.

3.1.3. Frontend

Para o frontend do proxecto, crearemos unha aplicación sinxela en Tomcat, empregando HTML. Facendo uso da funcionalidade que nos proporciona ERDDAP para xerar imaxes a partir dos datos, crearemos unha interface na que teñamos, por unha parte, un mapa cos datos que queiramos visualizar e, por outra parte, dous selectores, un para seleccionar o parámetro a consultar e outro para seleccionar o mes e ano.

3.2. Tecnoloxías empregadas

3.2.1. Anaconda

3.2.2. HARP

3.3. Desenvolvemento do sistema

3.3.1. Introducción

Para poder comezar a desenvolver o sistema, primeiro foi preciso realizar unha introdución á detección mediante satélite de diferentes datos e ó funcionamento dos satélites de TROPOMI. Para isto, foi de utilidade o seminario dispoñible en [4], de monitorización de alta resolución de NO_2 dende o espacio con TROPOMI. Nel, explícase o funcionamento básico dos satélites e como miden e procesan os datos. Tamén se explican os diferentes niveis de datos que existen, como mencionamos na introdución. Dipoñemos de 4 niveis principais de datos:

1. **Nivel 0 (L0):** estes datos son datos crus, tal e como se obteñen das medicións do satélite, a máxima resolución.
2. **Nivel 1 (L1):** son os datos de nivel L0, pero con referencias temporais e información dos sensores empregados para as medicións, coma coeficientes de calibración radiométrica e xeométrica.
3. **Nivel 2 (L2):** os datos de nivel L1, pero con variables xeofísicas derivadas na mesma resolución e ubicación que os datos de nivel 1.
4. **Nivel 3 (L3):** as variables obtidas dos datos de nivel L2, mapeadas en forma de cuadrículas espazo-temporais, completas e con consistencia.

Empregar datos de nivel L3 ten diversas vantaxes:

- Podemos dispoñer dun único arquivo por día (ou por mes, ou o período de tempo que nos interese). Polas características do satélite, o seu ciclo orbital é de 16 días, realizando 14 órbitas por día [6]. Polo tanto, para dous días distintos, é posible que un arquivo non cubra a totalidade da área de interese. Ó realizar as operacións de agregación dos datos, aseguramos ter un único arquivo por día, facendo a manipulación dos datos moito máis sinxela.
- Datos dispostos en cuadrículas uniformes. Coa agregación dos datos, podemos crear tamén unha cuadrícula para os mesmos, dispoñéndoo de forma regular no mapa ou área que nos interese estudar.

- Arquivos de menor tamaño. Os arquivos de nivel L2 conteñen unha maior cantidade de datos, unha gran parte dos cales pode non ser relevante para o noso sistema. O seu procesamento a nivel L3 permite eliminar estes datos e variables innecesarias, minguando o tamaño dos arquivos.
- Maior calidade nos datos. Nos diferentes manuais dos produtos, establécense diversos criterios de calidade asociados ós datos, que podemos empregar para descartar medicións de certas áreas pequenas do espazo en caso de que non teñan unha calidade suficiente. Desta forma, ó procesar os datos, podemos eliminar estas medicións de menor calidade e, como crearemos agregados por grandes períodos de tempo, evitaremos que queden vacíos.

O satélite SENTINEL 5P proporciona datos de seis parámetros moi relacionados coa contaminación atmosférica: ozono (O_3), tanto a columna troposférica (aquela que se forma en capas mais altas da atmosfera) coma a columna total; dióxido de nitróxeno (NO_2), tanto a columna troposférica coma a total; dióxido de xofre (SO_2), a columna total; monóxido de carbono (CO), a columna total; metano (CH_4), a columna total; e formaldehído ($HCHO$), a columna total. Tratar todos estes parámetros era difícil de abordar, polo que foi preciso descartar dous deles para non telos en conta. Baseándonos na información dispoñible en [7], decidimos elixir os seguintes: NO_2 , CO , SO_2 e O_3 . Escolléronse estes parámetros por seren os que máis problemas de saúde adoitan causar, ademais de por empregárense máis a miúdo nos diferentes mapas de contaminación.

3.4. Implementación do sistema

Capítulo 4

Probas

Capítulo 5

Conclusións e posibles ampliacións

Apéndice A

Manuais técnicos

En función do tipo de Traballo e metodoloxía empregada, o contido poderase dividir en varios documentos. En todo caso, neles incluírase toda a información precisa para aquelas persoas que se vaian encargar do desenvolvemento e/ou modificación do Sistema (por exemplo código fonte, recursos necesarios, operacións necesarias para modificacións e probas, posibles problemas, etc.). O código fonte poderase entregar en soporte informático en formatos PDF ou postscript.

Apéndice B

Manuais de usuario

Incluirán toda a información precisa para aquelas persoas que utilicen o Sistema: instalación, utilización, configuración, mensaxes de erro, etc. A documentación do usuario debe ser autocontida, é dicir, para o seu entendemento o usuario final non debe precisar da lectura doutro manual técnico.

Bibliografía

- [1] Data Products - Sentinel-5P Mission - Sentinel Online - Sentinel Online. Disponible en <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-5p/data-products>.
- [2] *ERDDAP - Home Page*. (s.f.). National Centers for Environmental Information (NCEI). Disponible en <https://www.ncei.noaa.gov/erddap/index.html>
- [3] Set Up Your Own ERDDAP. (s.f.). ERDDAP. Disponible en <https://coastwatch.pfeg.noaa.gov/erddap/download/setup.html>
- [4] *ARSET - High Resolution NO2 Monitoring From Space with TROPOMI*. NASA Applied Remote Sensing Training Program (ARSET). Disponible en <https://appliedsciences.nasa.gov/get-involved/training/english/arset-high-resolution-no2-monitoring-space-tropomi>
- [5] *Sentinel-5P L2*. (s.f.). Sentinel Hub. Disponible en <https://docs.sentinel-hub.com/api/latest/data/sentinel-5p-l2/>
- [6] *Types of air pollution. Which pollution is the most dangerous? — Air Quality Monitoring. Monitor in UK & Europe. Airly Data Platform and Monitors*. (s.f.). Air Quality Monitoring. Monitor in UK & Europe. Airly Data Platform and Monitors — Airly. <https://airly.org/en/the-most-dangerous-types-of-air-pollution/>
- [7] Documentation - OData. *Copernicus Dataspace* [en línea] [sen data] Disponible en: <https://documentation.dataspace.copernicus.eu/APIs/OData.html>