

# PrivaaaS - Anonymization Tools

*Eduardo Sakai, Tania Basso, Hebert Silva, Regina Moraes*

## Abstract

PrivaaaS is a set of libraries and tools developed in python that allow controlling and reducing data leakage in the context of Big Data processing and, consequently, protecting sensitive information that is processed by analytics algorithms.

In this version, you can apply data anonymization techniques on a set of tables according to anonymization policies. The PRIVAaaS provides generalization, suppression, masking, and encryption techniques. It is possible to apply the conjunction and disjunction process in the policies, in order to have a more or less restrictive data anonymization. The conjunction process applies an "AND" operation in the policies fields, anonymizing only the fields in the raw data which have a corresponding one in all policies. It means that, in this phase, the least restrictive policies were considered, hence maximizing data utility. The disjunction process applies an "OR" operation in the policies fields, which implies that all the fields must be anonymized according to their respective policies. This disjunction process results in the most restrictive anonymization, guaranteeing that the protection established by the policies are accomplished, performing the necessary anonymization to better protect the data.

In addition, the ARX library was integrated into the PrivaaaS to calculate the re-identification risk and, according to a risk threshold, to apply the k-anonymity algorithm to increase the anonymity level.

The python application allows csv and json files, both as input to the database and policies, as well as output format files. After calculating the risk and applying the k-anonymity algorithm by ARX the output dataset will be in csv format.

An initial user interface was implemented to manipulate the files and you can also use PrivaaaS as a service accessing the form with curl command line. Thus, the PrivaaaS can be used by other applications even if they were developed using different programming languages.

## Summary

<b>Technical Report - PrivaaaS</b>	<b>3</b>
Docker container	3
Programming languages.	3
Types of policies.	3
Supported file types.	4
Supported anonymization techniques.	4
Detailing Policy.	5
Implementation of conjunction and disjunction	9
Re-identification risk and k-anonymity	9
Install	11
Datasets	11
Policies	15
Settings	21
Conjunction	22
Disjunction	25
Download	27
Risk	29
<b>cURL for command line - PrivaaaS</b>	<b>34</b>
Sending datasets	34
Deleting datasets	34
Sending specific privacy policies	35
Sending global privacy policies	35
Deleting privacy policies	35
Generating anonymized datasets by conjunction process	35
Generating anonymized datasets by disjunction process	36
Downloading anonymized datasets	36
Deleting anonymized datasets	36
Calculating the re-identification risk and applying k-anonymity algorithm	36
Downloading anonymized datasets by k-anonymity algorithm	36
Deleting datasets anonymized using k-anonymity algorithm	37
Example of use	37
<b>References</b>	<b>38</b>

# Technical Report - PrivaaaS

## 1. Docker container

The Docker is an open source project which enables the packaging of applications or an entire environment inside a container. This container is portable to any other host that contains Docker installed.

PrivaaaS Anonymization Tools has been configured to work inside a Docker container, with all the dependencies necessary for its execution.

You can find more information about Docker on the official page of the project: <https://www.docker.com/>.

## 2. Programming languages.

The following programming languages can be used in the context of PrivaaaS:

- a. Python 3.6
  - i. Main Packages
    - 1. Flask 0.11.1
    - 2. Sqlalchemy 1.0.14
    - 3. Jinja2 2.8
- b. SQL
  - i. Sqlite
- c. HTML
  - i. Bootstrap 3.3.7
- d. CSS
  - i. Bootstrap 3.3.7

The database used in PrivaaaS has been implemented in Sqlalchemy, thus it is independent of the final database, being easily implemented to any database that uses the sql syntax. When a docker is instantiated in a host a Sqlite instance of the database is created.

## 3. Types of policies.

The privacy policies can be a global ou specific one:

a. Global Privacy Policy

- i. The global policy is based on recommendations from important institutes such as HIPAA (Health Insurance Portability and Accountability Act) and it is applied to all datasets that do not have a specific privacy policy.

b. Specific Privacy Policy

- i. A privacy policy defined by the data source owner and associated with a specific Dataset.

Limitations: this proof-of-concept version does not verify if all fields described in the privacy policy are the same in the dataset. This verification must be done by the data source owner.

#### **4. Supported file types.**

The following file types are available:

a. Datasets file types

- i. Csv
  - 1. Separators: “,”
- ii. Json
  - 1. Separators: “.”, “,”

b. Privacy policies file types

- i. Csv
  - 1. Separators: “,”
- ii. Json
  - 1. Separators: “.”, “,”
- iii. xml (Global privacy policy only)

Limitations: Privacy policy and its related dataset must have the same file type.

#### **5. Supported anonymization techniques.**

The following techniques are available:

- a. Generalization
- b. Suppression

- c. Masking
  - i. Faker Library
- d. Encryption
  - i. Flask-Hashing Library

## 6. Detailing Policy.

The privacy policy consists of a quintuple: name of the dataset, field name, type of privacy attribute, anonymization technique and hierarchy detail. These information are sufficient to apply the anonymization techniques on certain field names in the dataset.

The "DataSet" field refers to the name of the dataset associated with the privacy policy in question. Files of type csv or json are enabled.

The "FieldName" field indicates the name of the dataset field that should be anonymized.

The "PrivacyAttribute" field is the type of privacy attribute, it can assume the values: identifier ("IDENTIFIER"), quasi-identifier ("QUASI\_IDENTIFIER"), sensitive ("SENSITIVE"), and non-sensitive ("NON\_SENSITIVE").

The "AnonymizationTechnique" field specifies the anonymization technique that should be applied to the field specified in "FieldName". The accepted techniques are: suppression ("SUPPRESSION"), generalization ("GENERALIZATION"), masking ("MASK"), and encryption ("ENCRYPTION").

Finally, the "Details" field provides the specific details of each technique, such as, the size of the truncation, in the case of generalization or the type of encryption. More details can be found in the next section.

This quintuple structure can be provided by a json or csv file and serves either the global or the specific privacy policies. It is also possible to load xml-type files into global policies, following a different structure, as following.

Example of use parser parameters in three different types of files:

- a. xml files (Global anonymization policy only):

```
<AnonymizationOntology>
  <AnonymizationStrategy name="SafeHarborMethod">
    <Institution>HIPAA</Institution>
    <Rules>
      <Rule name="SafeHarborMethod-A">
        <FieldTable>Name</FieldTable>
        <FieldTableType>String</FieldTableType>
        <FieldTableClassification>KeyAttribute</FieldTableClassification>
        <Technique>Suppression</Technique>
        <Hierarchy>*</Hierarchy>
      </Rule>
    </Rules>
  </AnonymizationStrategy>
</AnonymizationOntology>
```

```

<Rule name="SafeHarborMethod-B">
  <FieldTable>BirthDate</FieldTable>
  <FieldTableType>Date</FieldTableType>
  <FieldTableClassification>QuasiIdentifier</FieldTableClassification>
  <Technique>Generalization</Technique>
  <Hierarchy>{"generalization_type": "truncate", "length": "4"}</Hierarchy>
</Rule>
<Rule name="SafeHarborMethod-J">
  <FieldTable>PostalCode</FieldTable>
  <FieldTableType>String</FieldTableType>
  <FieldTableClassification>QuasiIdentifier</FieldTableClassification>
  <Technique>Generalization</Technique>
  <Hierarchy>{"generalization_type": "truncate", "length": "2"}</Hierarchy>
</Rule>
</Rules>
</AnonymizationStrategy>
</AnonymizationOntology>

```

a. csv files:

```

DataSet;FieldName;PrivacyAttribute;AnonymizationTechnique;Details
dadosPessoais.csv;Name;IDENTIFIER;GENERALIZATION;{"generalization_type":
"truncate", "length": "1"}
dadosPessoais.csv;PostalCode;QUASI_IDENTIFIER;GENERALIZATION;{"generalization_type": "truncate", "length": "3"}
dadosPessoais.csv;Email;IDENTIFIER;MASK;{"lang": "pt_BR", "label_type": "email"}
dadosPessoais.csv;IDDocument;IDENTIFIER;ENCRYPTION;md5
dadosPessoais.csv;Salary;SENSITIVE;GENERALIZATION;{"generalization_type":
"rangeNumberToString", "hierarchy": ["Low=0-100", "Medium=100.01-500",
"High=500.01-"]}

```

b. json files:

```

[
  {
    "DataSet": "mock_data.json",
    "FieldName": "Name",
    "PrivacyAttribute": "IDENTIFIER",
    "AnonymizationTechnique": "MASK",
    "Details": {"lang": "en", "label_type": "name"}
  },
  {
    "DataSet": "mock_data.json",
    "FieldName": "email",
    "PrivacyAttribute": "QUASI_IDENTIFIER",
    "AnonymizationTechnique": "GENERALIZATION",
    "Details": {"generalization_type": "truncate", "length": "-5"}
  },
]

```

```

    {
      "DataSet": "mock_data.json",
      "FieldName": "last_name",
      "PrivacyAttribute": "IDENTIFIER",
      "AnonymizationTechnique": "ENCRYPTION",
      "Details": "md5"
    }
  ]

```

An important field of privacy policy is “Details”, where the details of the anonymization technique are defined. In the case of suppression, the characters defined in the “Details” field will overwrite all the values of the field specified in “FieldName”.

The generalization technique is the one that allows greater varieties of details. You can set truncation size or numeric or textual ranges. To do this, the “Details” field must contain a dictionary specifying the type of generalization and the length (in the case of truncate) or the hierarchy of the range.

In the case of encryption, the “Details” field must contain the type of encryption to apply.

Finally, in the case of applying the masking technique, the “Details” field must contain a dictionary, which in turn must contain the fields: the language (lang) and the field type (label\_type).

Usage examples for defining anonymization techniques are following.

a. Suppression

- i. Field value is exchanged for a string passed in details.

b. Generalization

i. Truncate

1. “Details” = {‘generalization\_type’: ‘truncate’, ‘length’: ‘1’}

ii. Range

1. Number to Text

a. Example:

- i. “Details”: {“generalization\_type”: “rangeNumberToString”, “hierarchy”: {“group1”: (x,y), “group2”: (u,v)}}
- ii. “Details”: {“generalization\_type”: “rangeNumberToString”, “hierarchy”: [“Crianca=0-12”, “Adolescente=13-19”, “Adulto=20-60”, “Idoso=60-”]}

- iii. `"Details":{"generalization_type":"rangeNumberToString","hierarchy":["Low=0-100","Medium=100.01-500","High=500.01-"]}`

## 2. Text to Text

### a. Example:

- i. `"Details":{"generalization_type":"rangeStringToString", "hierarchy": {"group1":(x,y,z), "group2":(s,u,v)}}`
- ii. `"Details":{"generalization_type":"rangeStringToString", "hierarchy":["São Paulo=Palmeiras-Corinthians-São Paulo-Santos","Rio de Janeiro=Flamengo-Vasco-Fluminense-Botafogo","Rio Grande do Sul=Grêmio-Internacional","Minas Gerais=Cruzeiro-Atlético","Bahia=Bahia"]}`

### c. Encryption

#### i. Supported encryption

- 1. md5
- 2. sha1
- 3. sha224
- 4. sha256
- 5. sha384
- 6. sha512

#### ii. Example of use (json file):

- 1. `"Details": "md5"`
- 2. `"Details": "sha512"`

### d. Masking

#### i. Using python Faker package

Faker is a python package that generates fake data.

##### 1. Supported label types

- a. In [\[1\]](#) you can check all types of supported labels.

##### 2. Supported languages

- a. In [\[2\]](#) you can check all supported languages.

##### 3. Example of use:



- a. "Details": { "lang": "en", "label\_type": "email" }
- b. "Details": { "lang": "pt\_BR", "label\_type": "name" }

## 7. Implementation of conjunction and disjunction

The implementation of conjunction and disjunction processes is explained below:

- a. Conjunction
  - i. The conjunction applies an AND rule to all Datasets, looking for common field names for all Datasets loaded in the system.
- b. Disjunction
  - i. The disjunction applies an OR rule to Datasets, i.e., if there is an anonymization technique associated with a field name in the privacy policy files (global or specific). This technique will be applied to the records in that field.

## 8. Re-identification risk and k-anonymity

The ARX anonymization tool provide techniques and models to perform data anonymization and methods for calculating re-identification risk and allows the user make the balance with data utility [3].

We integrated the methods available in ARX to build a solution for data anonymization based on a risk threshold.

To perform k-anonymity based on risk threshold, you need to edit the "Risk config" file. In this file you inform the name of fields and the type of this data in terms of data classification (number "1" for non-sensitive fields; number "2" for identifiers; number "3" for quasi identifiers; number "4" for sensitive data). The ARX anonymization requires that you inform at least 1 quasi-identifier. Then you can use "SR" to suppress digits from Right to left, "SL" to suppress digits from left to Right, "DT" to create ranges of date and "AG" to create ranges of age.

```

ZIPCODE;SR
DATETIME;1
MIN;1
MAX;1
COUNT;1
SUM;1
Rmax;0.01

```

WARNING: The last single line, "Rmax" represent the Risk accepted to output data set in decimal format. The terms must be separated by semicolon.

- a. ARX k-anonymity
  - i. Using jar with a python class.
- b. Example of risk configuration.

```
Name;2
PostalCode;SL
Email;2
Company;4
Phone;2
BrithDate;DT
Salary;1
Rmax;0.6
```

After calculating the risk, a verification is performed: if this risk is higher than the threshold established in the "Risk Config" file, the value of k (from k-anonymity) is increased and k-anonymity is applied again with the new value of k, in order to increase the anonymity level and, consequently, reduce the risk. This is performed successively, until the re-identification risk is equal to or lower than the threshold.

WARNING: All values at the fields (attributes) customized must have contemplated in the created hierarchy. The columns must be separated by semicolon.

To perform generalization, the component provides some hierarchy files. They are in the folder "hierarchy" (priva/development/arx-poc/hierarchy) and define ranges for education (e.g., primary school, high school, undergraduate, etc.), native countries (e.g., asia, north america, europe, etc.), occupation, age, among others. An example of customized hierarchy is given below:

```
Married-civ-spouse;spouse present;*
Divorced;spouse not present;*
Never-married;spouse not present;*
Separated;spouse not present;*
Widowed;spouse not present;*
Married-spouse-absent;spouse not present;*
Married-AF-spouse;spouse present;*
```

This example is for generalizing marital status. The first level of hierarchy will be equal to original input. The second level of hierarchy will be "Spouse present" or "Spouse not present". The last hierarchy level is suppressed by "\*\*".

If you need to customize your own hierarchy, you need to create a new csv file named "custom.csv" into the "hierarchy" folder. After creating the desired hierarchy, change the "risk config" file, placing "CT" in front of the field (attribute) to be generalized.

## User Interface Manual - PrivaaaS

This user manual is intended to guide the user in the process of installing and using PrivaaaS Anonymization Tools.

### 1. Install

To run PrivaaaS, you only need install Docker, and after that the process is fully automated. The installation of docker is beyond the scope of this document.

Clone the project or download it from git project:

```
$ git clone https://github.com/eubr-bigsea/privaaas.git
```

In linux or mac just execute the two files in project root folder: Docker-build.sh and Docker-run.sh. Your sudo password will be requested.

1. Build the Docker image

```
$ source Docker-build.sh
```

2. Run Docker

```
$ source Docker-run.sh
```

3. Visit the site:

<http://localhost:5000>

In Windows you need to run the following commands:

1. Build the Docker image

```
$ docker build -f Dockerfile -t privaaas .
```

2. Run Docker

```
$ docker run -d -p 5000:5000 privaaas
```

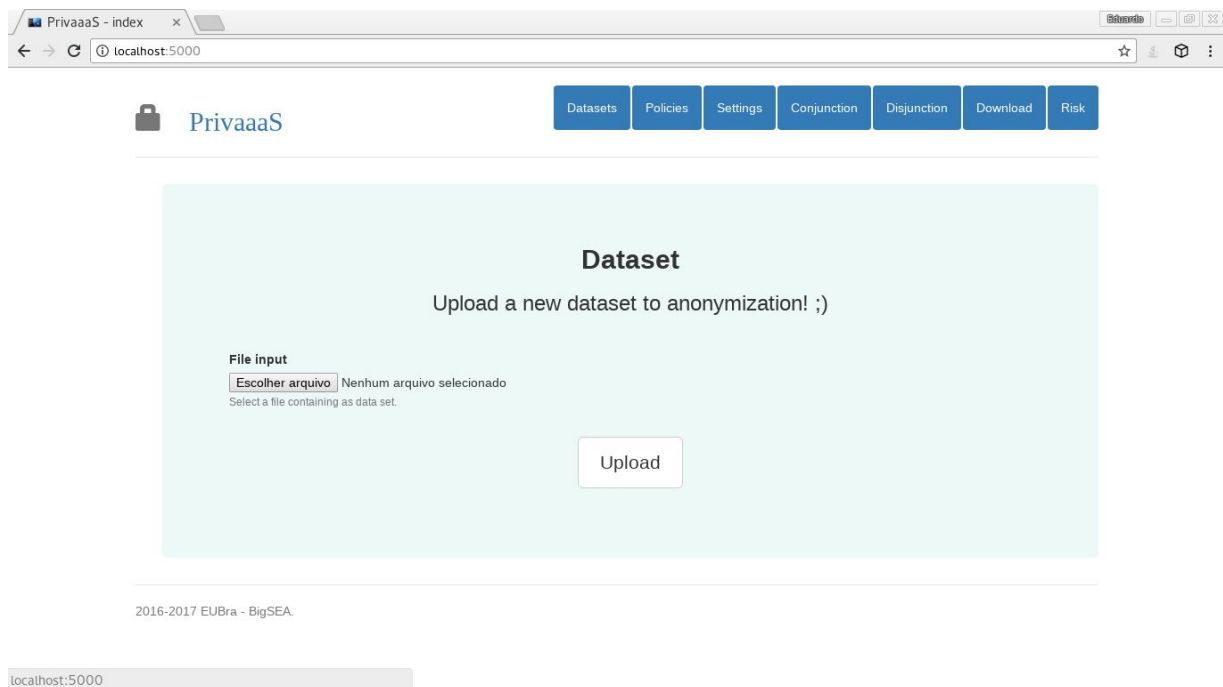
3. Visit the site:

<http://localhost:5000>

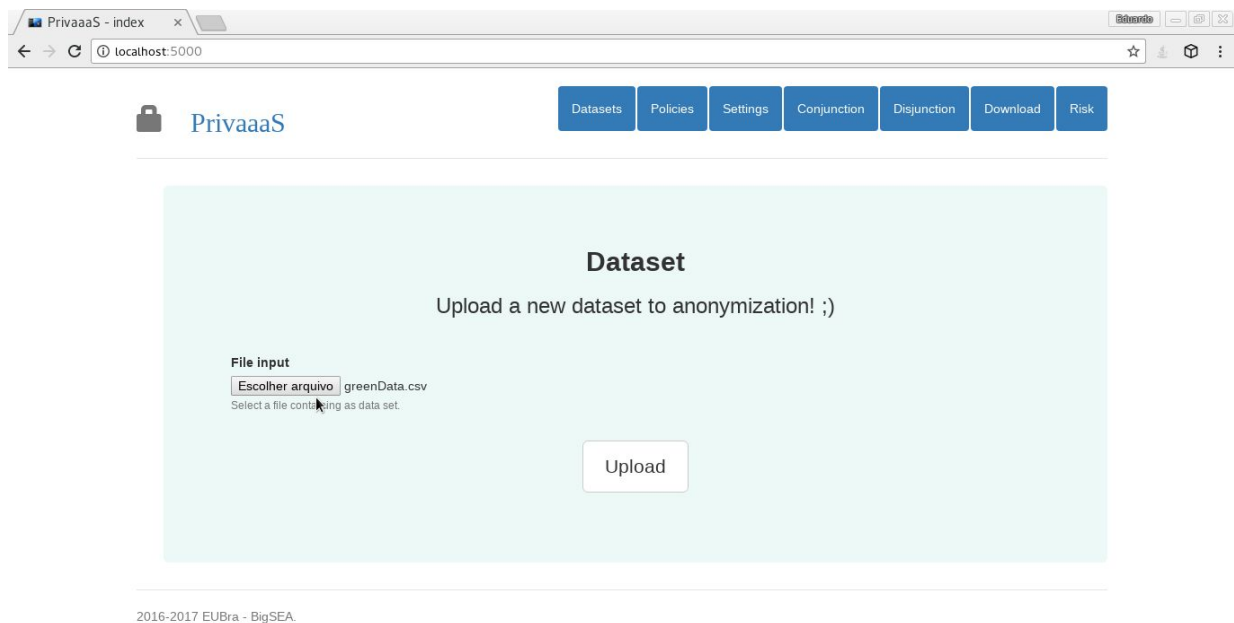
### 2. Datasets

The supported input dataset file types are csv and json.

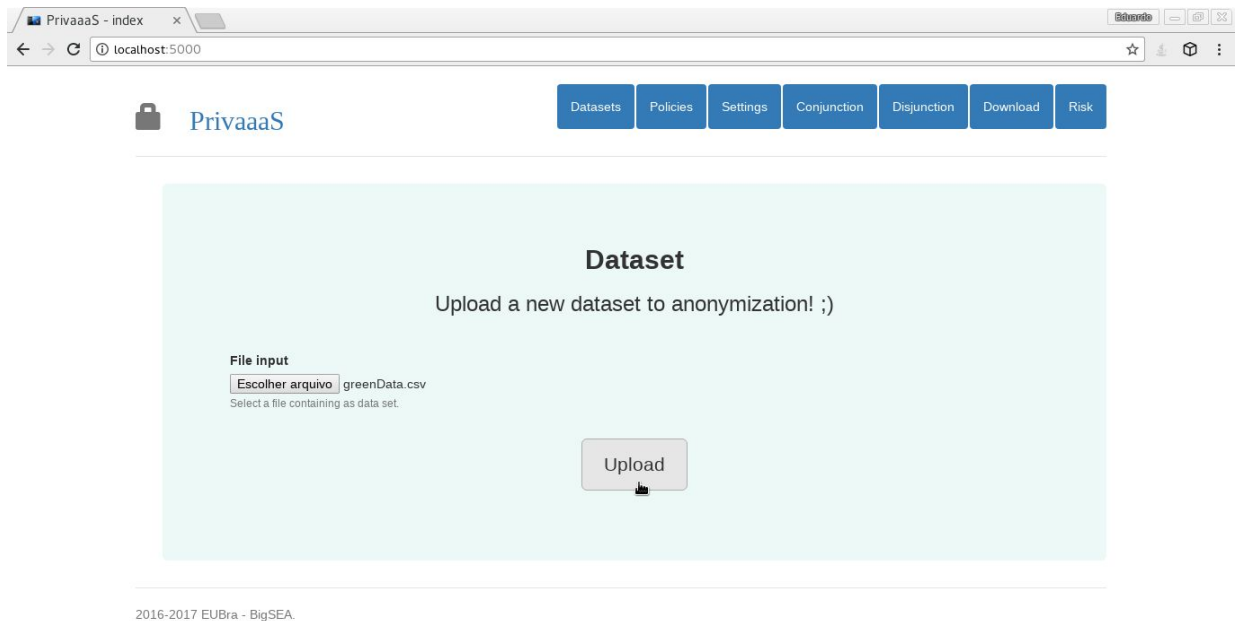
The home page corresponds to the Datasets menu. On this page you can add datasets and delete them if needed.



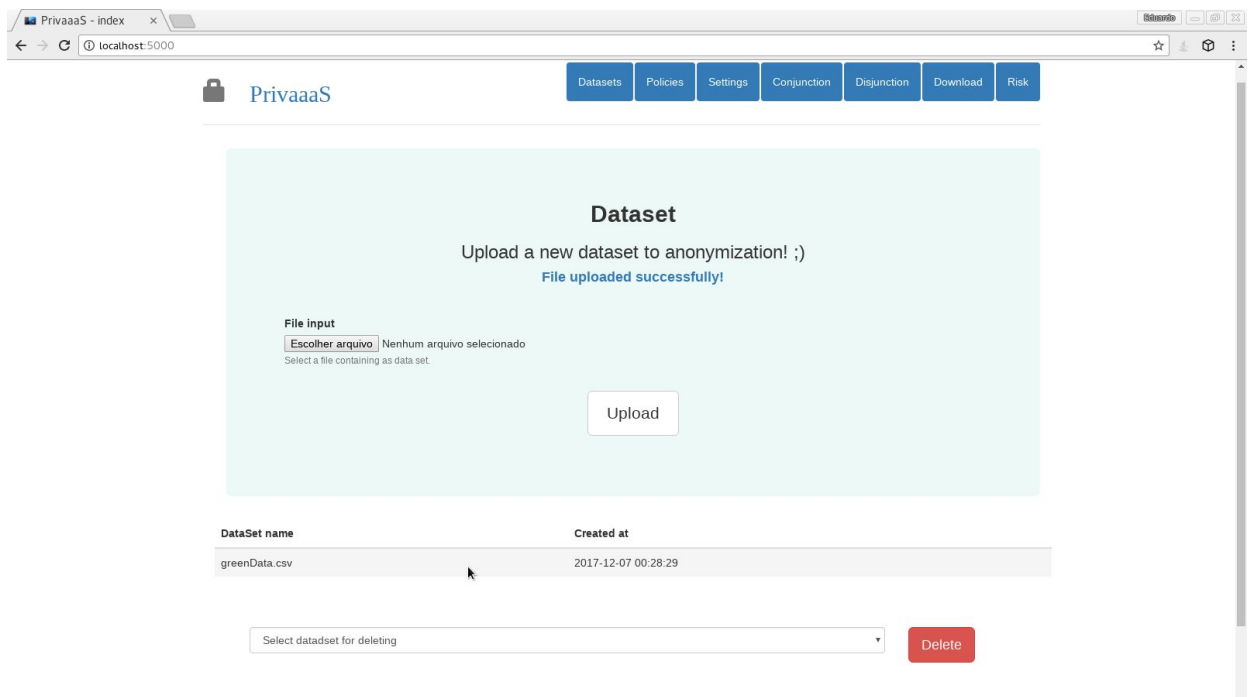
To upload a new dataset to PrivaaaS system, just click "Select file" and choose the file you want to upload.



Then click “Upload”.

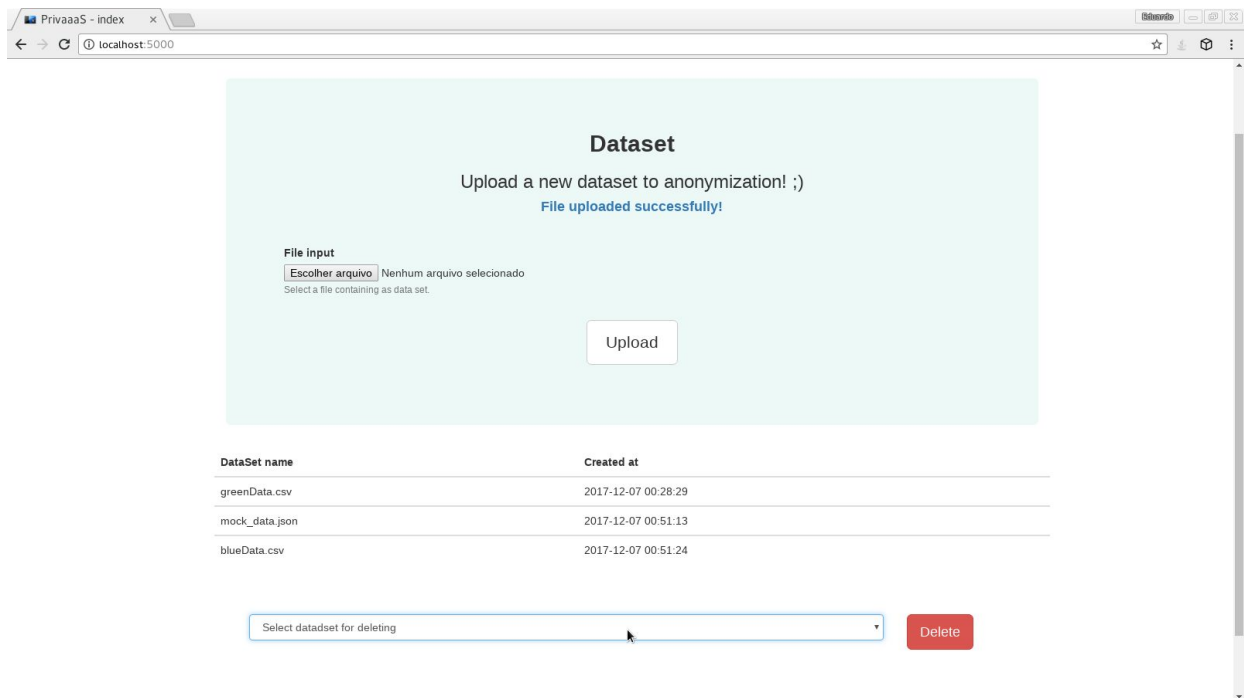


At the bottom of the screen you can see the datasets that have already been loaded into the PrivaaaS.

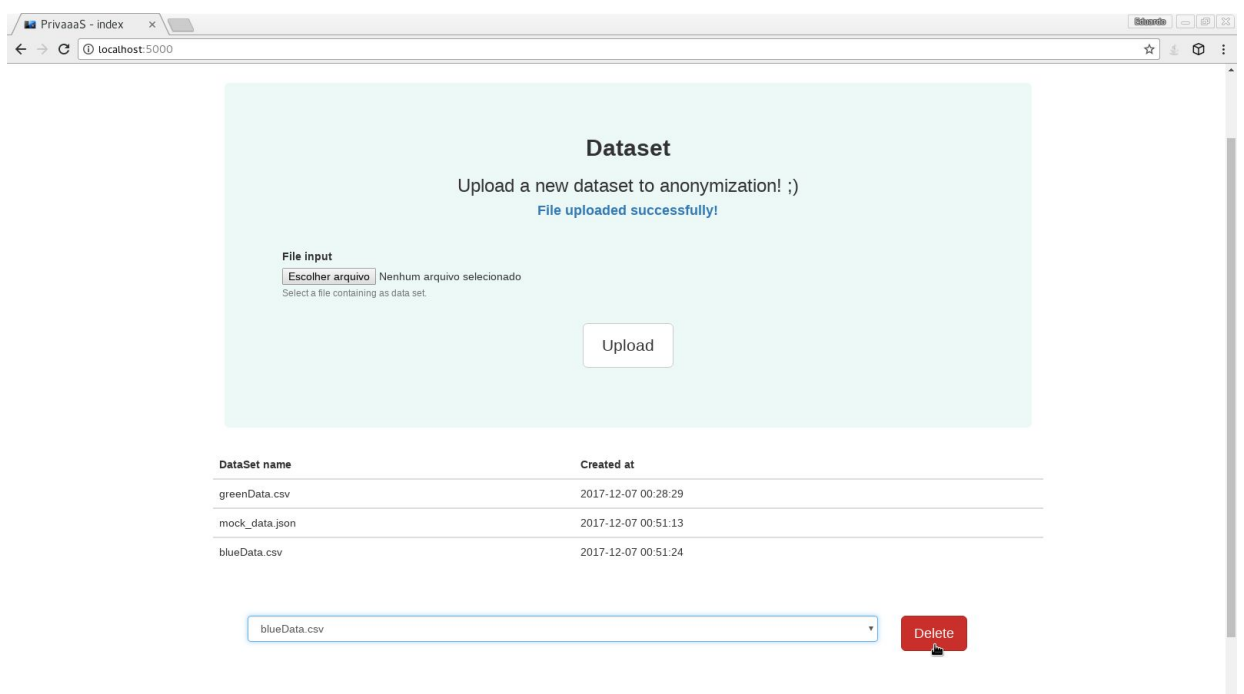


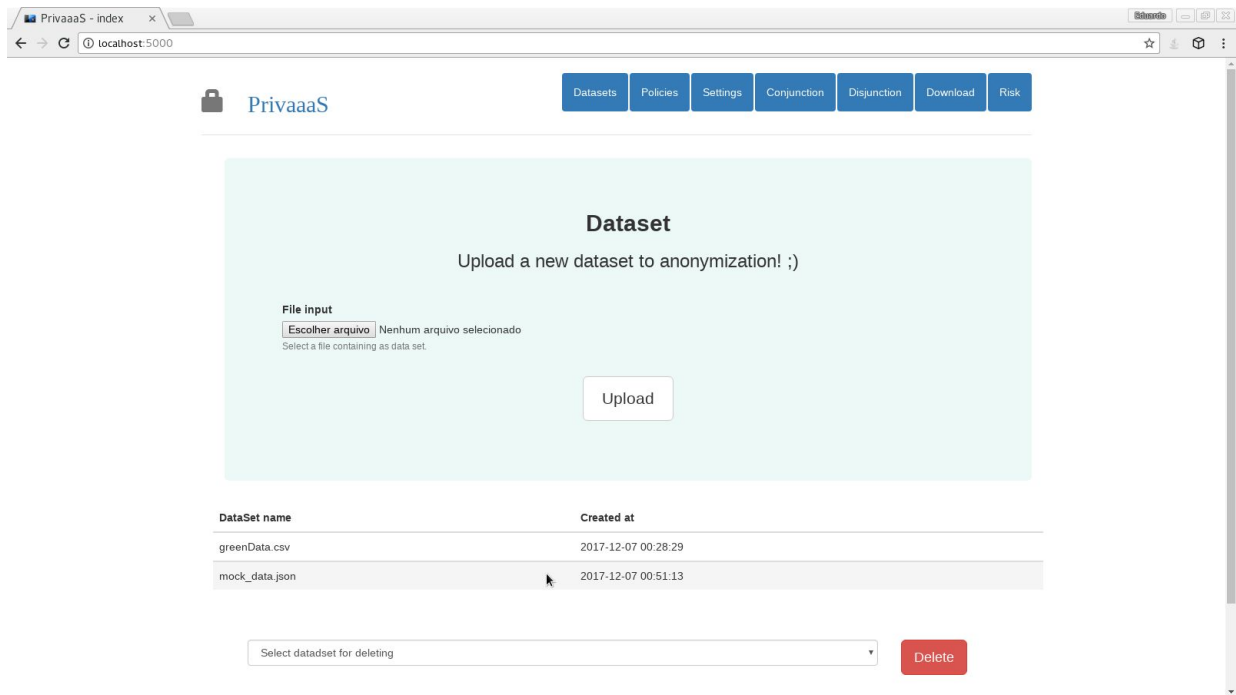
You can repeat the process if you want to upload more datasets.

If you want to delete a dataset, just click on "Select dataset for deleting" and select the file.



After selecting the dataset, just click on "Delete".





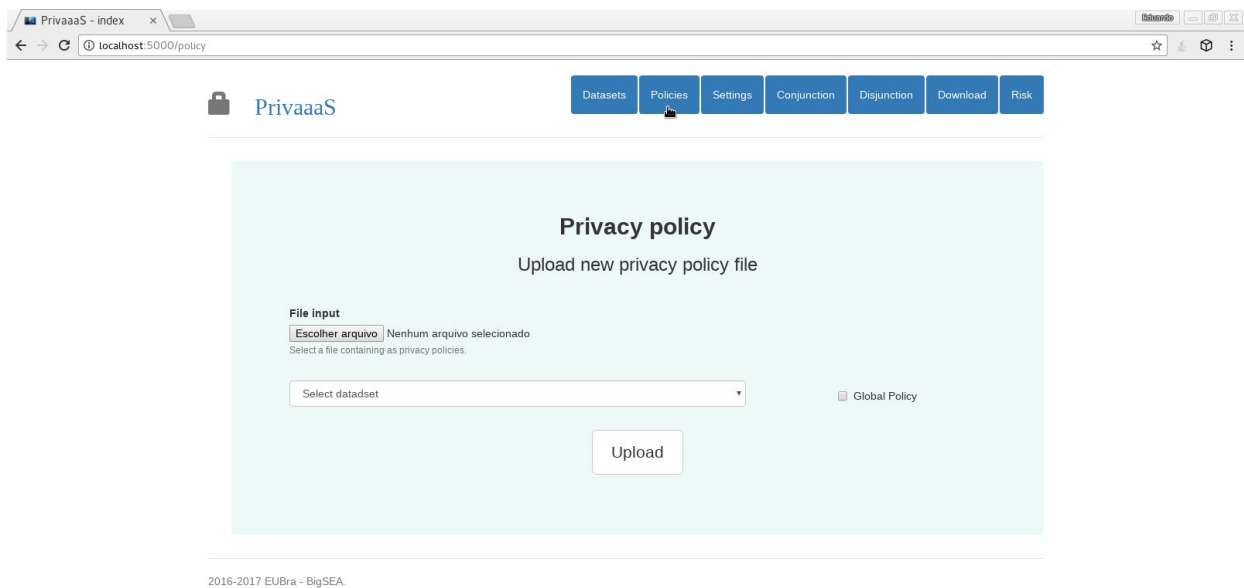
**WARNING:** If you delete a dataset that has a specific privacy policy linked to the dataset, the privacy policy also will be deleted.

**WARNING:** In this version the system does not ask for confirmation to delete the file.

### 3. Policies

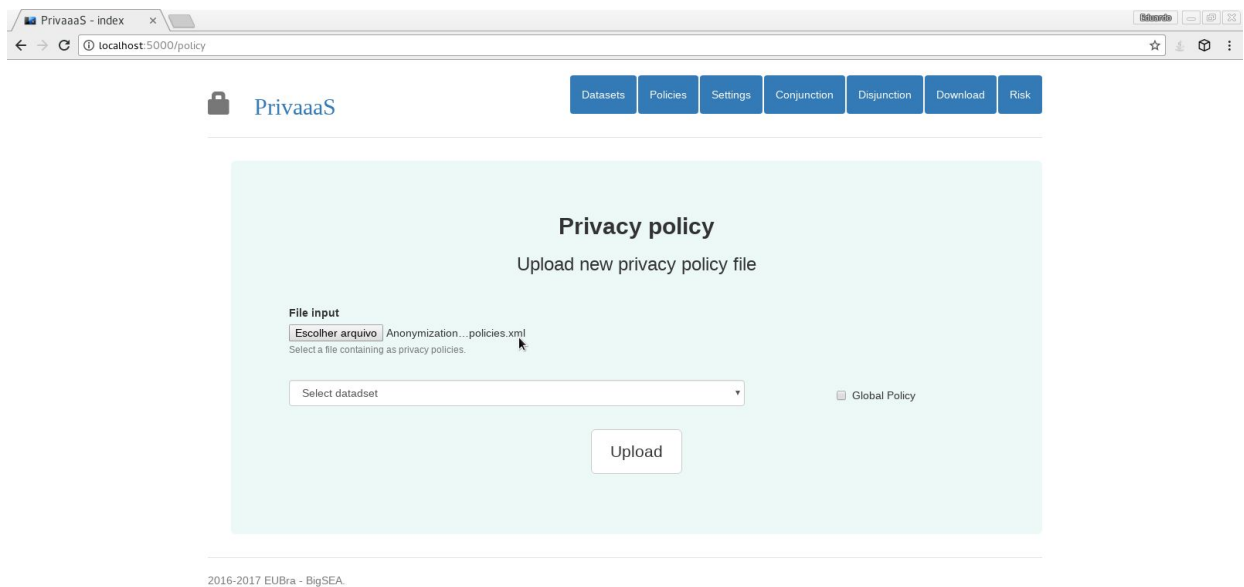
The type of privacy policy file should match the type of dataset file. Input privacy policy file can be csv or json type files. Global privacy policy can be defined also in XML format.

By clicking on the "Policies" menu, the loading screen of privacy policy file opens.



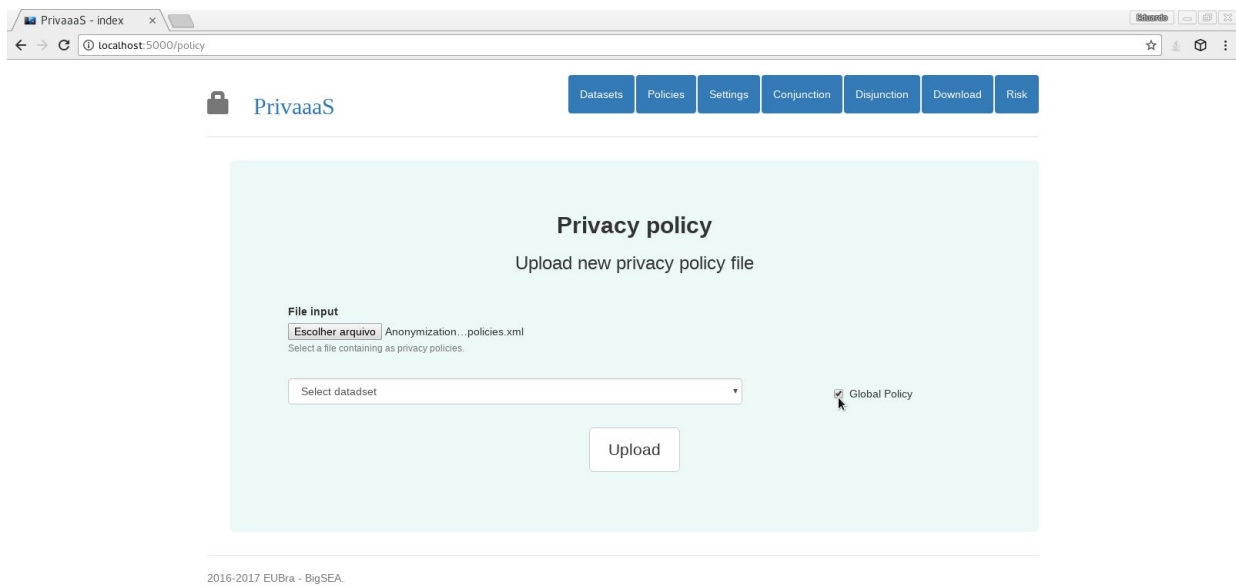
localhost:5000/policy

To upload a new privacy policy, you must click "Select File" and select the file.

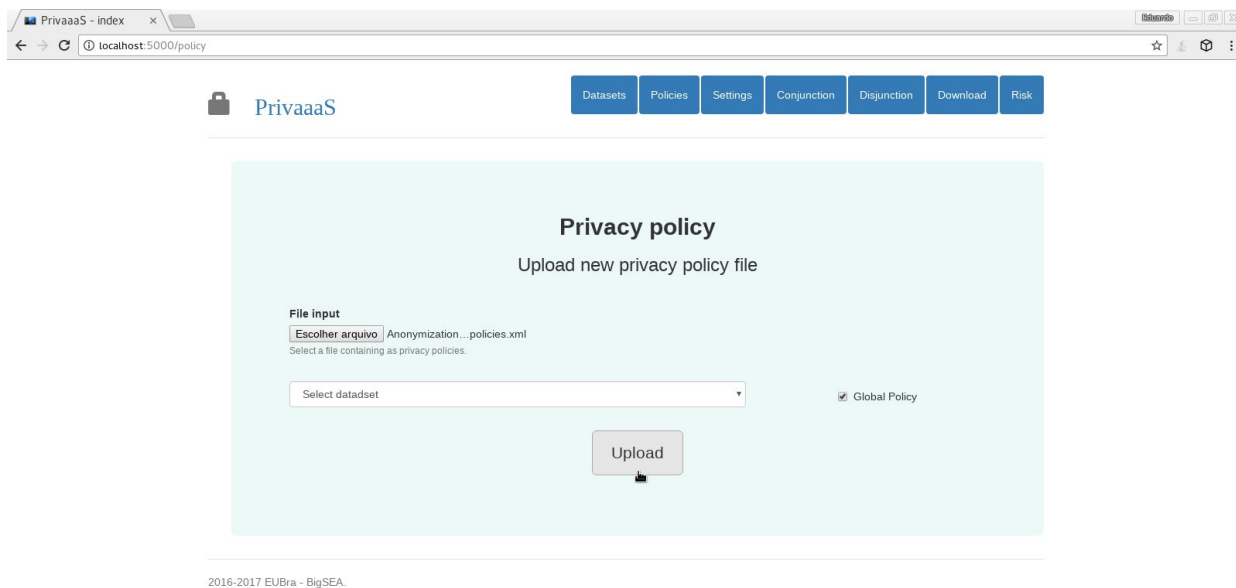


If the privacy policy to be add if its all datasets (global policy), you must check this option, as shown below.

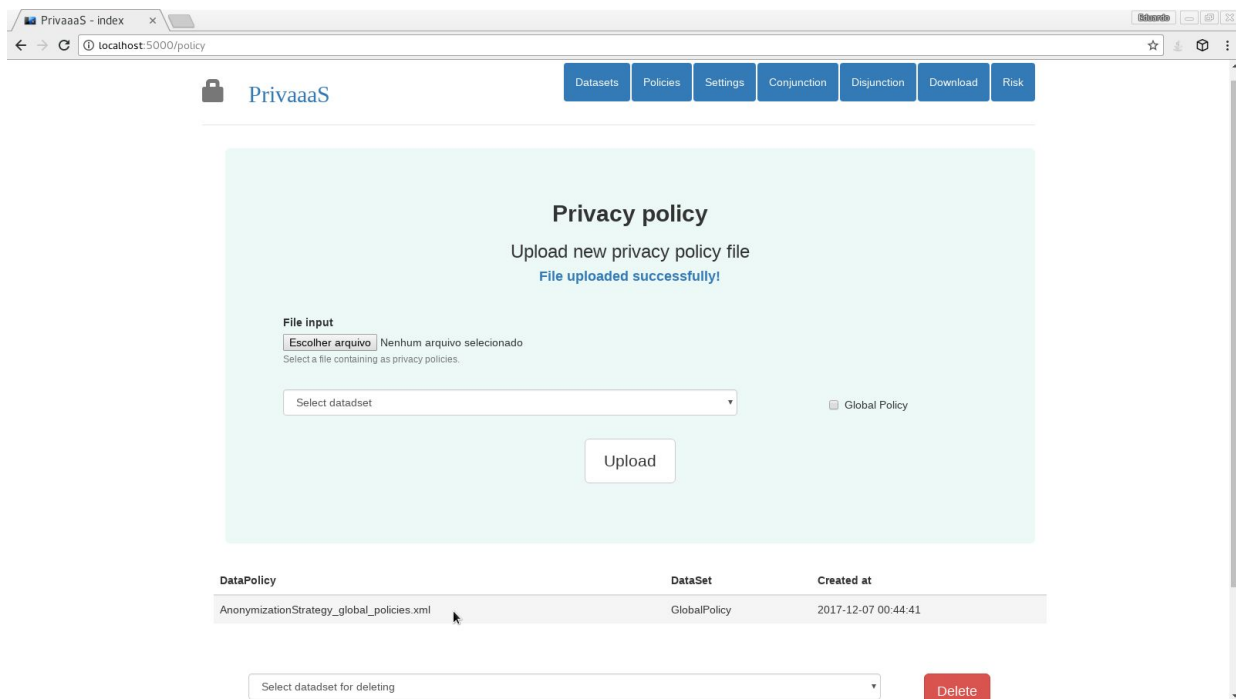




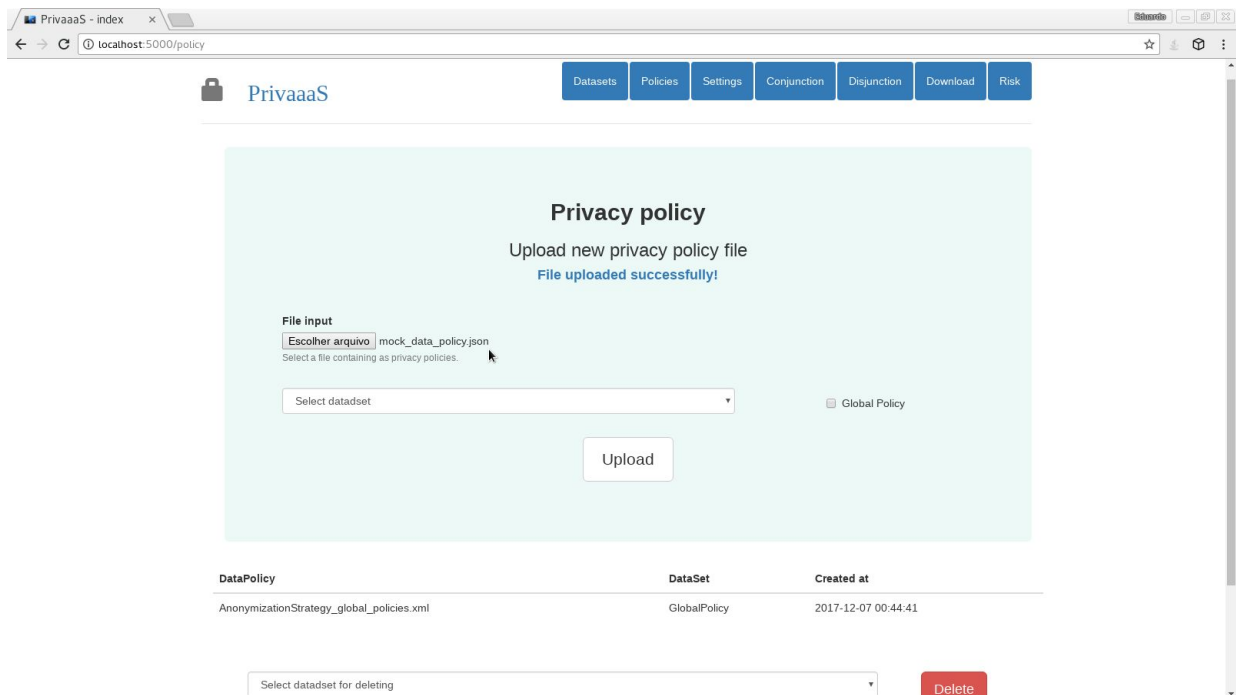
And then click "Upload".

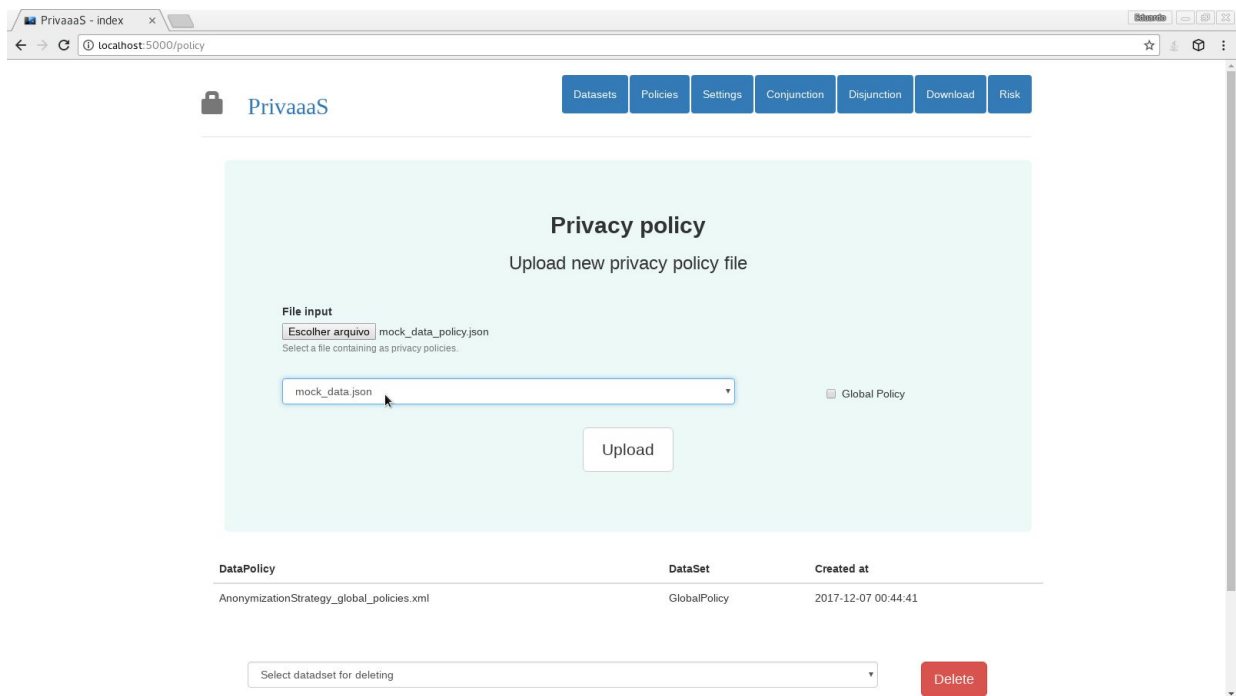


Following, you can view the privacy policies that have already been loaded into the system.

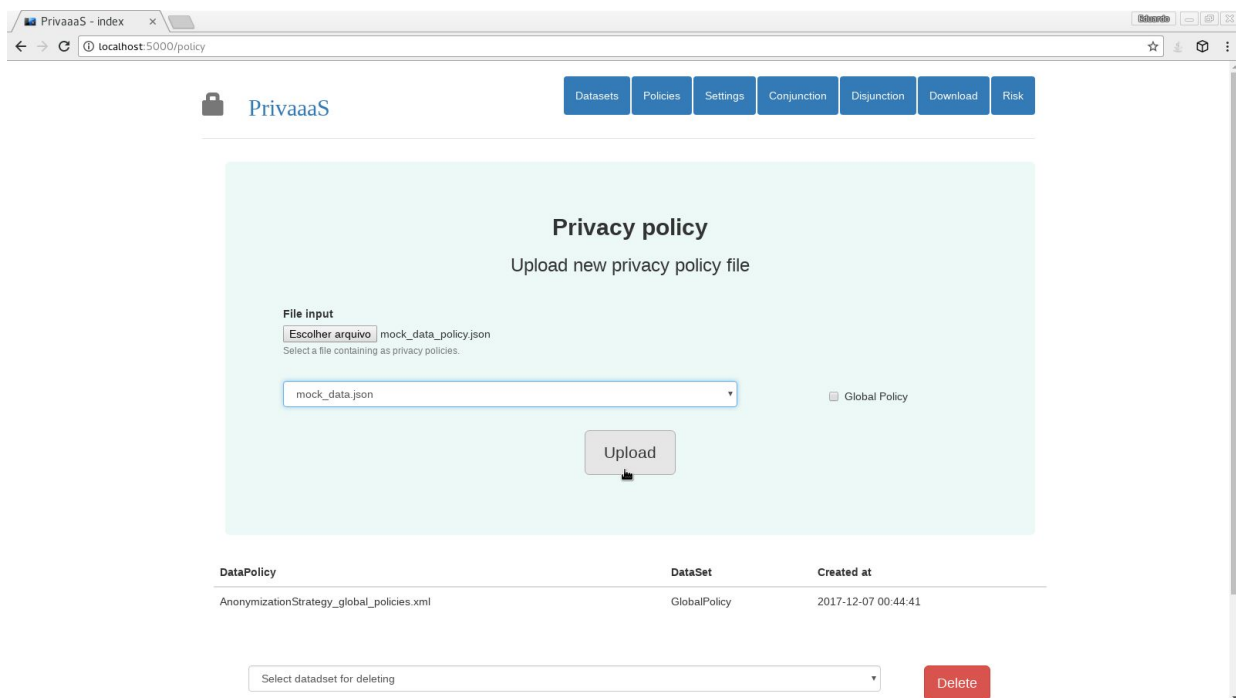


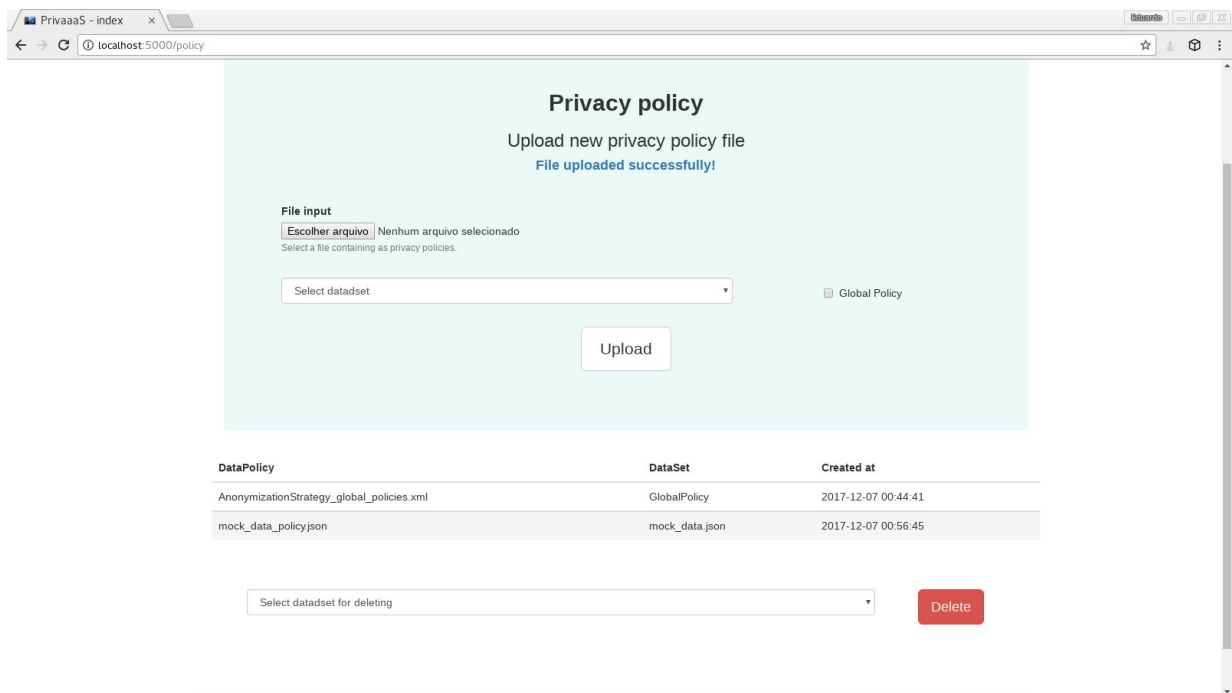
If the privacy policy is specific to a dataset, you must select the corresponding dataset by clicking "Select dataset".



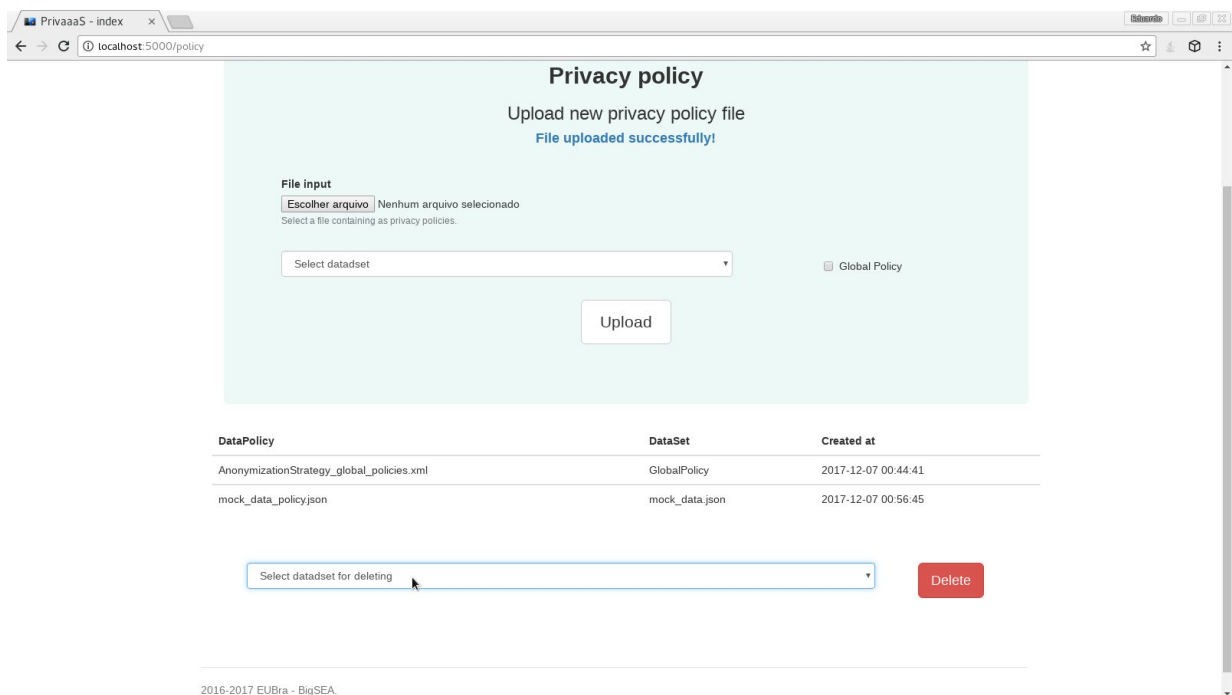


And then click "Upload".





If you want to delete a privacy policy, simply click "select dataset for deleting" and select the file. After selecting the dataset, just click on "Delete" button.



**WARNING:** In this version the system does not ask for confirmation to delete the file.

## 4. Settings


You can not change a field name of a global privacy policy. You only can change the field name of the specific privacy policies.

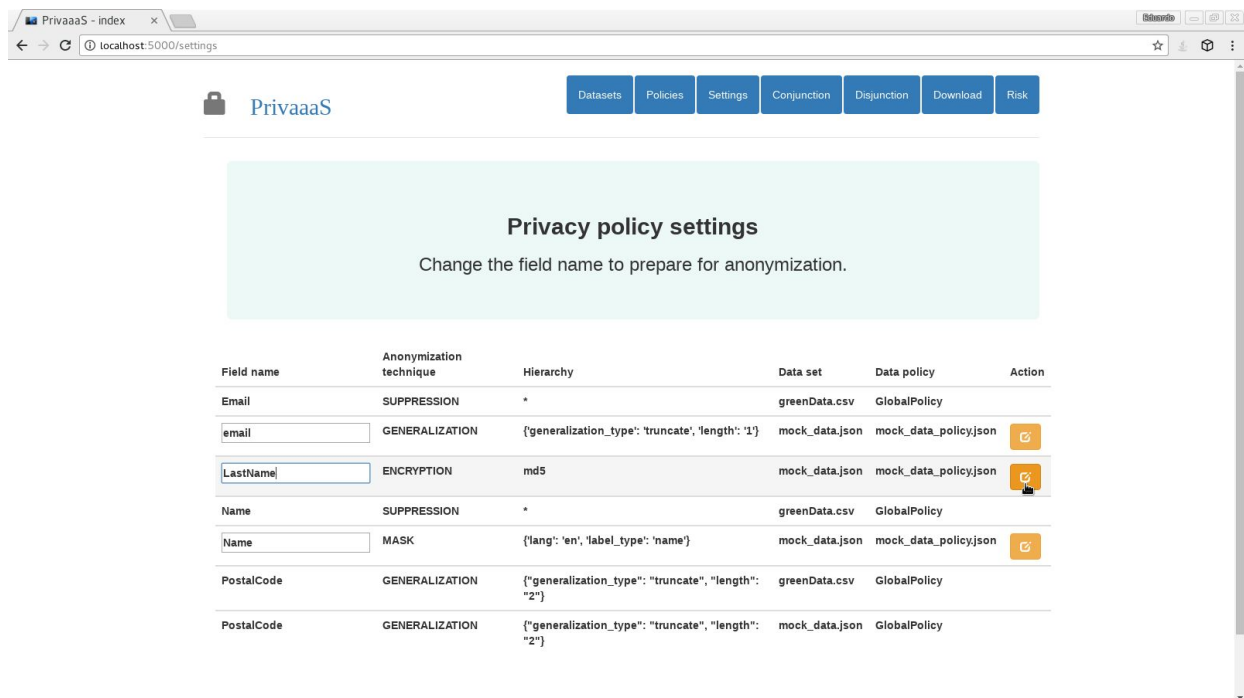
In the Settings menu you can change the field name of the specific policy files to a specific dataset.

The screenshot shows the PrivaaaS web application interface. The top navigation bar includes links for Datasets, Policies, Settings (active), Conjunction, Disjunction, Download, and Risk. The main heading is "Privacy policy settings" with a subtext "Change the field name to prepare for anonymization." Below this is a table with columns: Field name, Anonymization technique, Hierarchy, Data set, Data policy, and Action.

Field name	Anonymization technique	Hierarchy	Data set	Data policy	Action
Email	SUPPRESSION	*	greenData.csv	GlobalPolicy	
<input type="text" value="email"/>	GENERALIZATION	{'generalization_type': 'truncate', 'length': '1'}	mock_data.json	mock_data_policy.json	
<input type="text" value="last_name"/>	ENCRYPTION	md5	mock_data.json	mock_data_policy.json	
Name	SUPPRESSION	*	greenData.csv	GlobalPolicy	
<input type="text" value="Name"/>	MASK	{'lang': 'en', 'label_type': 'name'}	mock_data.json	mock_data_policy.json	
PostalCode	GENERALIZATION	{'generalization_type': 'truncate', 'length': '2'}	greenData.csv	GlobalPolicy	
PostalCode	GENERALIZATION	{'generalization_type': 'truncate', 'length': '2'}	mock_data.json	GlobalPolicy	

This screenshot is identical to the one above, but with the 'last\_name' row in the table highlighted. The 'Field name' column for this row shows the text 'last\_name' with a cursor at the end, indicating it is selected for editing.

To save the change to a field name, just change the desired field and click on  button corresponding to the field name (i.e., which is in the same line).



WARNING: This action change the field name in the input dataset.

## 5. Conjunction

The conjunction process refers to the Anonymization 1 stage (this stage refers to anonymization in ETL process, when data are loaded in big data platforms [4]). To perform the conjunction, all datasets must contain at least one common field name to be accepted in the conjunction condition. The technique of anonymization used in case of disagreement among the techniques is the most restrictive in this ranking: 1- suppression, 2 - encryption, 3 - masking, and 4 - generalization.

By accessing the menu, a table displays the names of fields that satisfy the conjunction condition, i.e., the common field names for all datasets.

**PrivaaaS Conjunction**  
Applying the conjunction rule.

Privacy policy generated

Details	DataSet	AnonymizationTechnique	PrivacyAttribute	FieldName
*	greenData.csv	SUPPRESSION	IDENTIFIER	Name
*	mock_data.json	SUPPRESSION	IDENTIFIER	Name
{'generalization_type': 'truncate', 'length': 2}	greenData.csv	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
{'generalization_type': 'truncate', 'length': 2}	mock_data.json	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
*	greenData.csv	SUPPRESSION	IDENTIFIER	Email
*	mock_data.json	SUPPRESSION	IDENTIFIER	email

Anonymize

To generate the corresponding anonymized datasets, simply click "Anonymize" button.

**PrivaaaS Conjunction**  
Applying the conjunction rule.

Privacy policy generated

Details	DataSet	AnonymizationTechnique	PrivacyAttribute	FieldName
*	greenData.csv	SUPPRESSION	IDENTIFIER	Name
*	mock_data.json	SUPPRESSION	IDENTIFIER	Name
{'generalization_type': 'truncate', 'length': 2}	greenData.csv	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
{'generalization_type': 'truncate', 'length': 2}	mock_data.json	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
*	greenData.csv	SUPPRESSION	IDENTIFIER	Email
*	mock_data.json	SUPPRESSION	IDENTIFIER	email

Anonymize

Following, you can see the field names that were anonymized.

The screenshot shows the PrivaaS web interface at localhost:5000/privacy\_and. At the top, there is a table with 5 columns: {generalization\_type: 'truncate', 'length': 2}, greenData.csv, GENERALIZATION, QUASI\_IDENTIFIER, and PostalCode. Below this table is a green 'Anonymize' button. Under the button, the text 'Anonymized fieldname' is followed by a list of field names: "Name", "Name", "PostalCode", "PostalCode", "Email", and "email". Below this list, the text 'Dataset: greenData.csv' is followed by a table with 6 columns: Name, Telephone, PostalCode, Gender, Email, and Age. The table contains 6 rows of data.

Name	Telephone	PostalCode	Gender	Email	Age
*	(63) 1164-2810	92***	Male	*	105
*	(62) 6864-8700	83***	Male	*	1
*	(21) 6932-2194	85***	Male	*	105
*	(61) 2513-0365	23***	Female	*	1
*	(43) 2735-2826	23***	Male	*	105
*	(65) 1912-7209	06***	Male	*	105

All the anonymized data sets are displayed in sequence, one after another.

The screenshot shows the PrivaaS web interface at localhost:5000/privacy\_and. It displays the same 'Anonymized fieldname' list as the previous screenshot. Below it, the text 'Dataset: greenData.csv' is followed by a larger table with 6 columns: Name, Telephone, PostalCode, Gender, Email, and Age. This table contains 20 rows of data.

Name	Telephone	PostalCode	Gender	Email	Age
*	(63) 1164-2810	92***	Male	*	105
*	(62) 6864-8700	83***	Male	*	1
*	(21) 6932-2194	85***	Male	*	105
*	(61) 2513-0365	23***	Female	*	1
*	(43) 2735-2826	23***	Male	*	105
*	(65) 1912-7209	06***	Male	*	105
*	(24) 8286-1280	60***	Female	*	1
*	(61) 5363-9038	33***	Male	*	105
*	(63) 6429-0457	89***	Female	*	1
*	(40) 7952-5452	03***	Male	*	1
*	(84) 5766-5236	92***	Male	*	105
*	(81) 0559-0161	97***	Female	*	105
*	(85) 7197-1108	01***	Female	*	105
*	(33) 4919-0397	98***	Female	*	1
*	(71) 3802-0087	85***	Male	*	105
*	(45) 8507-5950	07***	Female	*	1

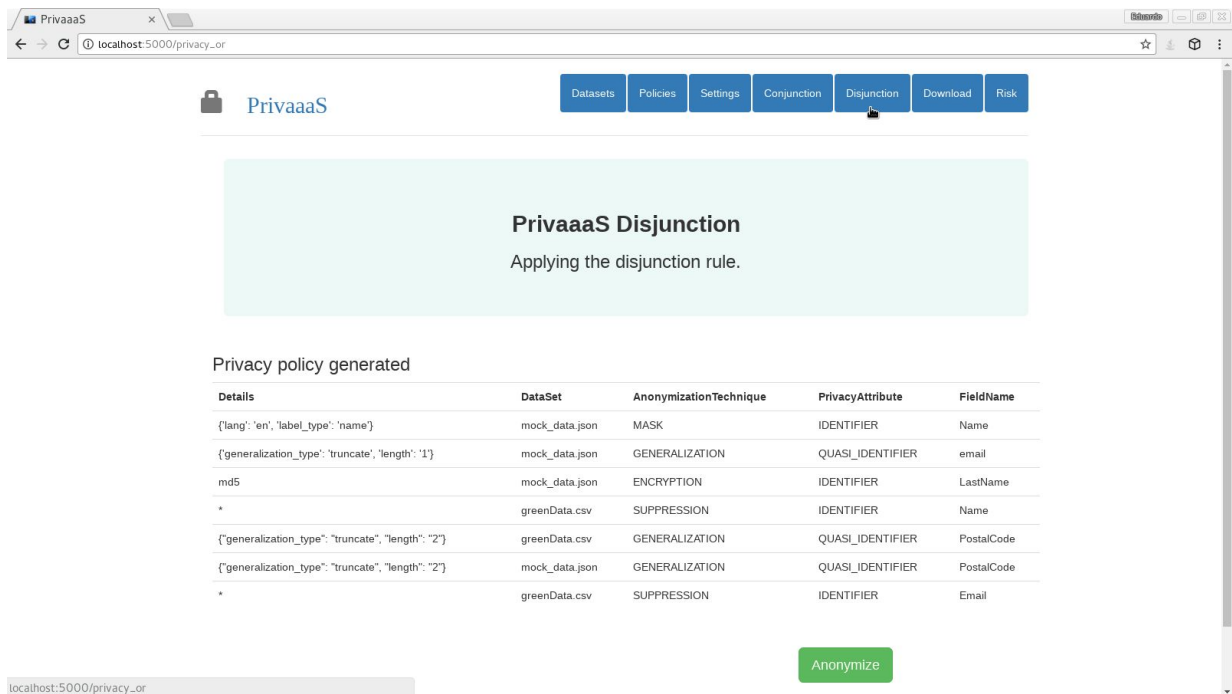
The datasets generated are available for downloading by clicking the "Download" menu.



## 6. Disjunction

The disjunction process refers to the Anonymization 2 stage (this stage refers to anonymization of the data provided by the analytics algorithm, i.e., the intermediary data sets [4]). To perform the disjunction process It is enough that there is a policy (global or specific) stating that the name of the field must be anonymized. The anonymization technique used is the same as described by the policies.

The disjunction menu shows a table that displays the names of fields that satisfy the condition of disjunction, i.e., it lists the fields that have some anonymization technique associated to them.



The screenshot shows the PrivaaaS Disjunction interface. At the top, there is a navigation bar with buttons for Datasets, Policies, Settings, Conjunction, Disjunction (highlighted), Download, and Risk. Below the navigation bar, the title "PrivaaaS Disjunction" is displayed, followed by the subtitle "Applying the disjunction rule." Below this, a section titled "Privacy policy generated" contains a table with the following data:

Details	DataSet	AnonymizationTechnique	PrivacyAttribute	FieldName
{'lang': 'en', 'label_type': 'name'}	mock_data.json	MASK	IDENTIFIER	Name
{'generalization_type': 'truncate', 'length': '1'}	mock_data.json	GENERALIZATION	QUASI_IDENTIFIER	email
md5	mock_data.json	ENCRYPTION	IDENTIFIER	LastName
*	greenData.csv	SUPPRESSION	IDENTIFIER	Name
{'generalization_type': 'truncate', 'length': '2'}	greenData.csv	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
{'generalization_type': 'truncate', 'length': '2'}	mock_data.json	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
*	greenData.csv	SUPPRESSION	IDENTIFIER	Email

At the bottom right of the interface, there is a green button labeled "Anonymize".

To generate the corresponding anonymized datasets, just click on "Anonymize" button.

PrivaaaS Disjunction  
Applying the disjunction rule.

Privacy policy generated

Details	DataSet	AnonymizationTechnique	PrivacyAttribute	FieldName
{'lang': 'en', 'label_type': 'name'}	mock_data.json	MASK	IDENTIFIER	Name
{'generalization_type': 'truncate', 'length': '1'}	mock_data.json	GENERALIZATION	QUASI_IDENTIFIER	email
md5	mock_data.json	ENCRYPTION	IDENTIFIER	LastName
*	greenData.csv	SUPPRESSION	IDENTIFIER	Name
{'generalization_type': 'truncate', 'length': '2'}	greenData.csv	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
{'generalization_type': 'truncate', 'length': '2'}	mock_data.json	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
*	greenData.csv	SUPPRESSION	IDENTIFIER	Email

Anonymize

2016-2017 EUBra - BigSEA

The anonymized data sets are displayed in sequence, one after another.

{'generalization\_type': 'truncate', 'length': '2'}

mock_data.json	GENERALIZATION	QUASI_IDENTIFIER	PostalCode
greenData.csv	SUPPRESSION	IDENTIFIER	Email

Anonymize

Anonymized fieldname

"Name"	"email"	"LastName"	"Name"
"PostalCode"	"PostalCode"	"Email"	

Dataset: greenData.csv

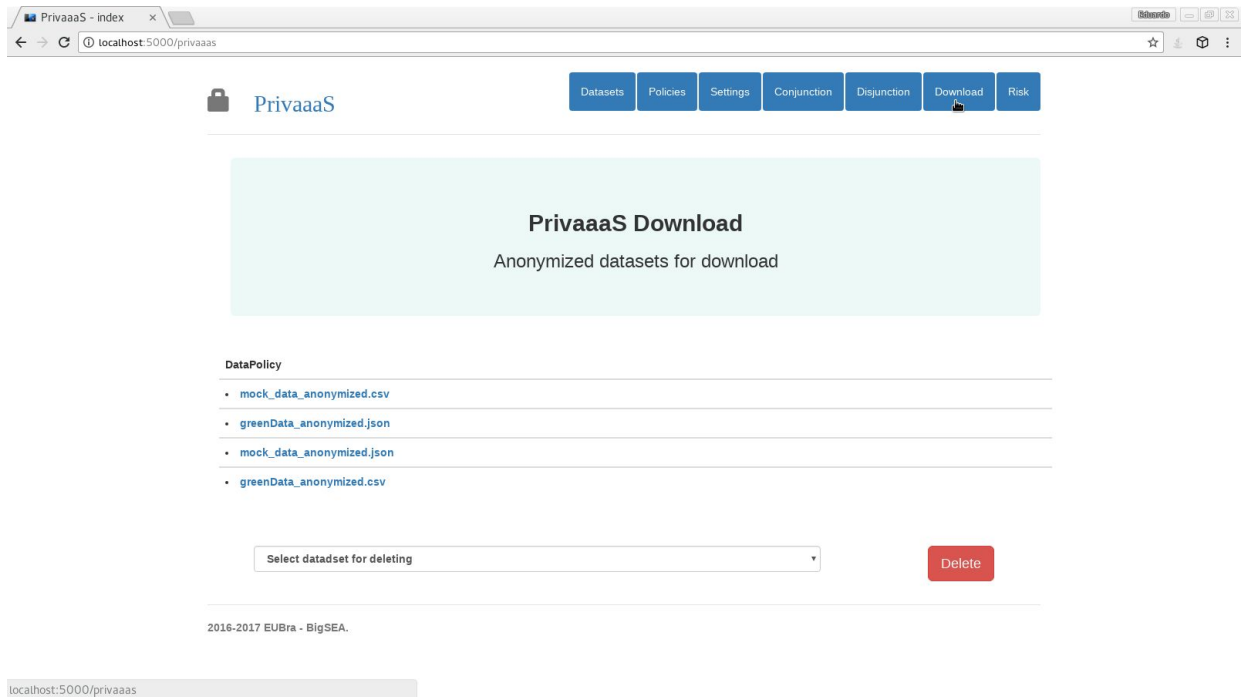
Name	Telephone	PostalCode	Gender	Email	Age
*	(63) 1164-2810	92***	Male	*	105
*	(62) 6864-8700	83***	Male	*	1
*	(21) 6932-2194	85***	Male	*	105
*	(61) 2513-0365	23***	Female	*	1
*	(43) 2735-2826	23***	Male	*	105
*	(65) 1912-7209	06***	Male	*	105
*	(24) 8286-1280	60***	Female	*	1
*	(61) 5363-9038	33***	Male	*	105

The datasets generated are available for downloading by clicking the "PrivaaaS" menu.

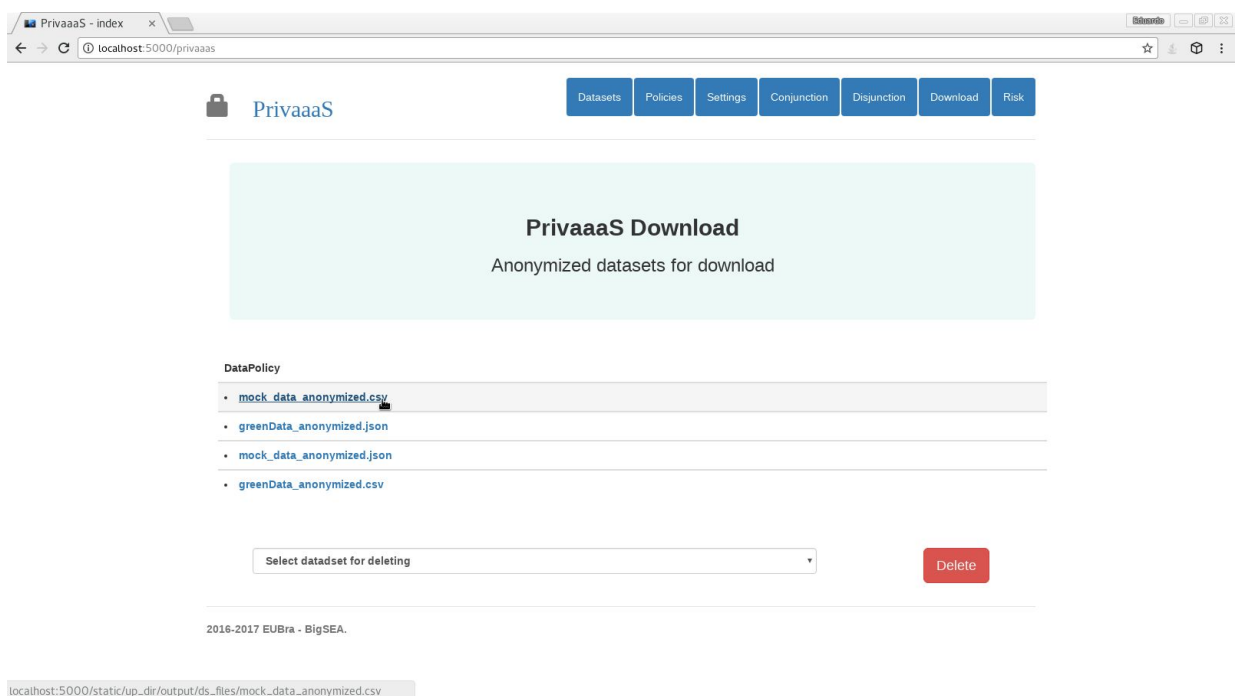
## 7. Download

When clicking the "Anonymize" button, either at the conjunction or at the disjunction, anonymized files will be generated, both in json and csv format, which are enabled for downloading.

In "Download" menu you will find for downloading the anonymized datasets.



To download a specific dataset, if it is the csv type, simply click on the file browser. It will ask where to save the file and click "Save" to start the download.



If the desired dataset is in json format, by clicking on the desired file its contents will be displayed on the screen. To save the file, you must click with the right mouse button and select "Save as", the browser will ask where to save and start to download it.

PrivaaaS - index

localhost:5000/privaaaS

PrivaaaS

Datasets Policies Settings Conjunction Disjunction Download Risk

### PrivaaaS Download

Anonymized datasets for download

DataPolicy

- mock\_data\_anonymized.csv
- greenData\_anonymized.json
- mock\_data\_anonymized.json
- greenData\_anonymized.csv

Select dataset for deleting

Delete

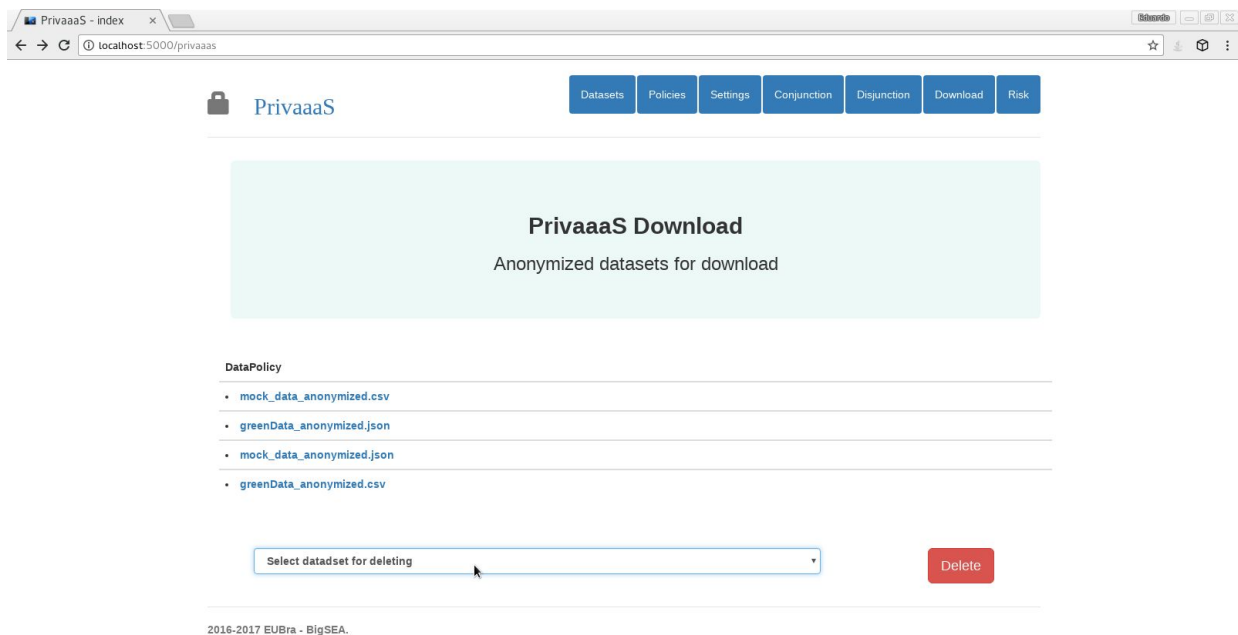
2016-2017 EUBra - BigSEA.

localhost:5000/static/up\_dir/output/ds\_files/mock\_data\_anonymized.json

localhost:5000/static/up\_dir/output/ds\_files/mock\_data\_anonymized.json

```
[
  {
    "Name": "Michelle Copeland",
    "id": 1,
    "PostalCode": "71*****",
    "job": "Help Desk Technician",
    "gender": "Female",
    "education": "Maharshi Dayanand Saraswati University Ajmer",
    "email": "m*****",
    "salary": "$4.53",
    "LastName": "00978ffc73cea5b0da6d988d36736374"
  },
  {
    "Name": "Corey Henderson",
    "id": 2,
    "PostalCode": "75*****",
    "job": "Media Manager IV",
    "gender": "Female",
    "education": "National Taiwan College of Physical Education and Sports",
    "email": "c*****",
    "salary": "$5.51",
    "LastName": "57018edc1773d7dd8e57fe450f504873"
  },
  {
    "Name": "Tonya Raymond",
    "id": 3,
    "PostalCode": "75*****",
    "job": "Administrative Officer",
    "gender": "Male",
    "education": "Adi University of Science and Technology",
    "email": "b*****",
    "salary": "$6.34",
    "LastName": "7a3b015f4c73bf6ab77d02dffe3420a"
  },
  {
    "Name": "Leslie Roman",
    "id": 4,
    "PostalCode": "22*****",
    "job": "Database Administrator I",
    "gender": "Male",
    "education": "Trinity University",
    "email": "h*****",
    "salary": "$4.06",
    "LastName": "221b40a5ea9862d0d7fb4c09796d9ab1"
  },
  {
    "Name": "Willie Sutton",
    "id": 5,
    "PostalCode": "13*****",
    "job": "Recruiting Manager",
    "gender": "Male",
    "education": "Universidad \"Adolfo Iba\u00f1ez\"",
    "email": "h*****",
    "salary": "$2.12",
    "LastName": "21c41505720be1eb4e1e70f203d53774"
  }
]
```

If you want to delete an anonymized dataset, just click on "Select dataset for deleting" and select the file.

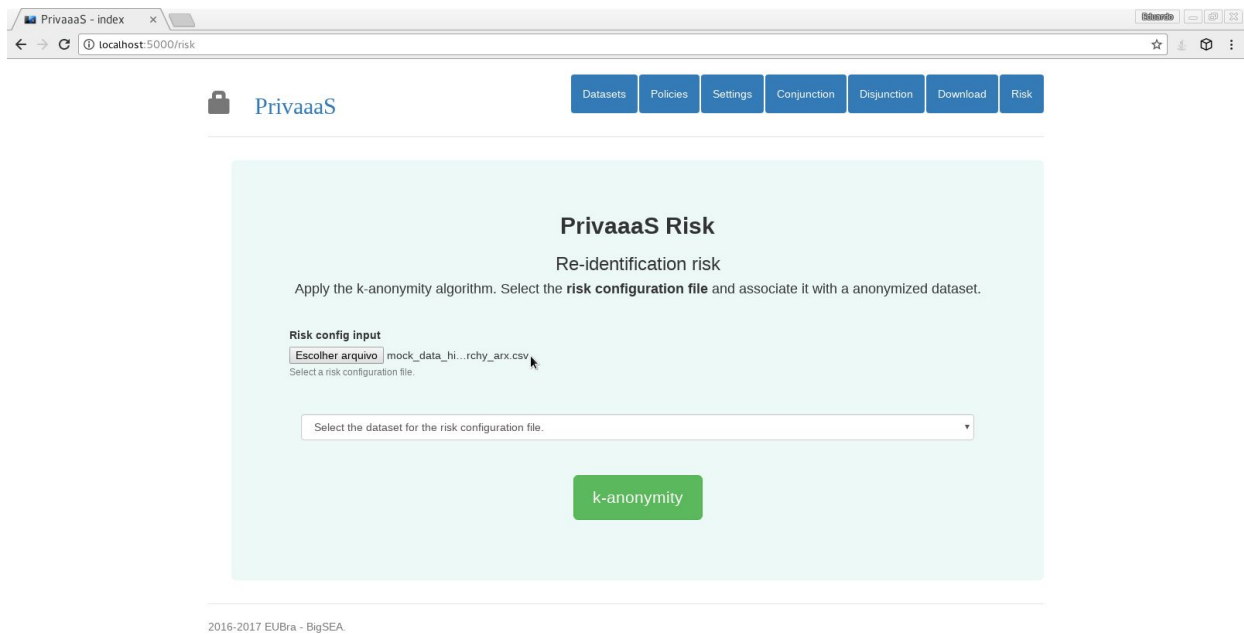
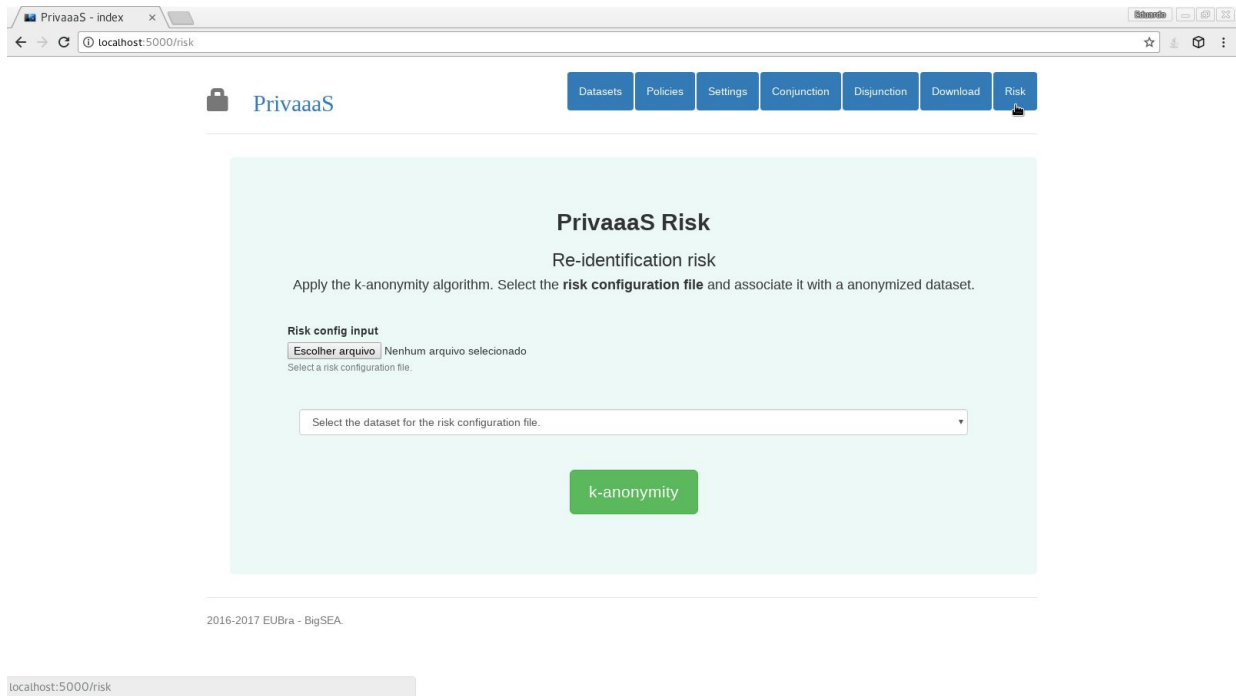


After selecting the dataset, just click on "Delete" button.

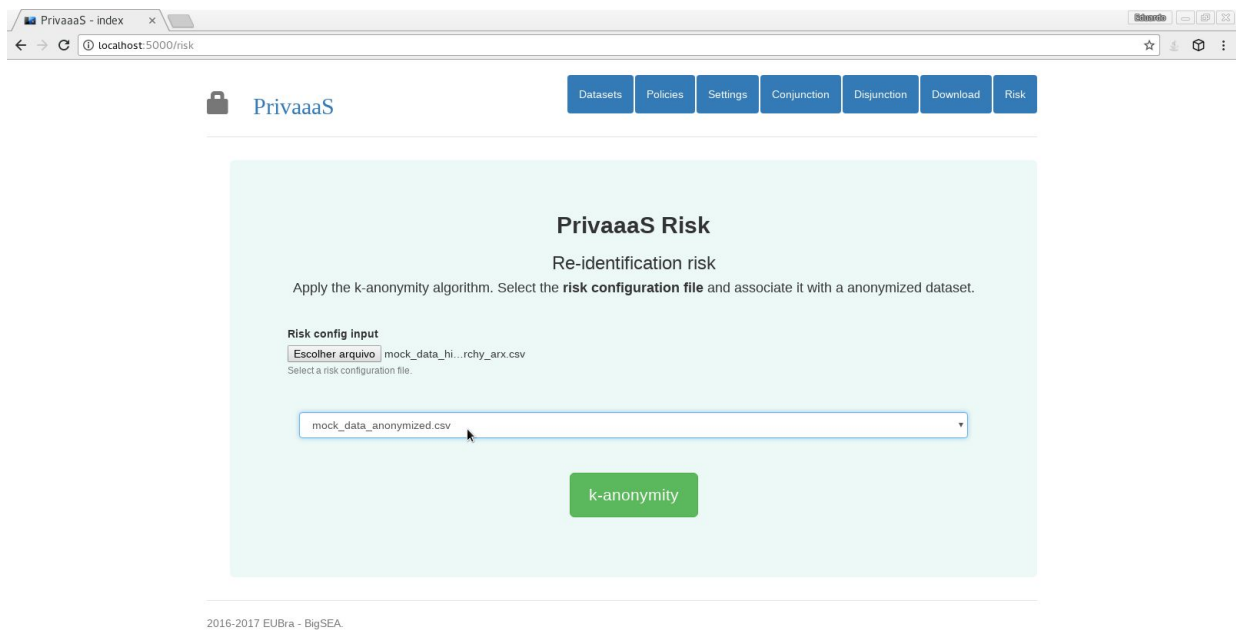
**WARNING:** In this version the system does not ask for confirmation to delete the file.

## 8. Risk

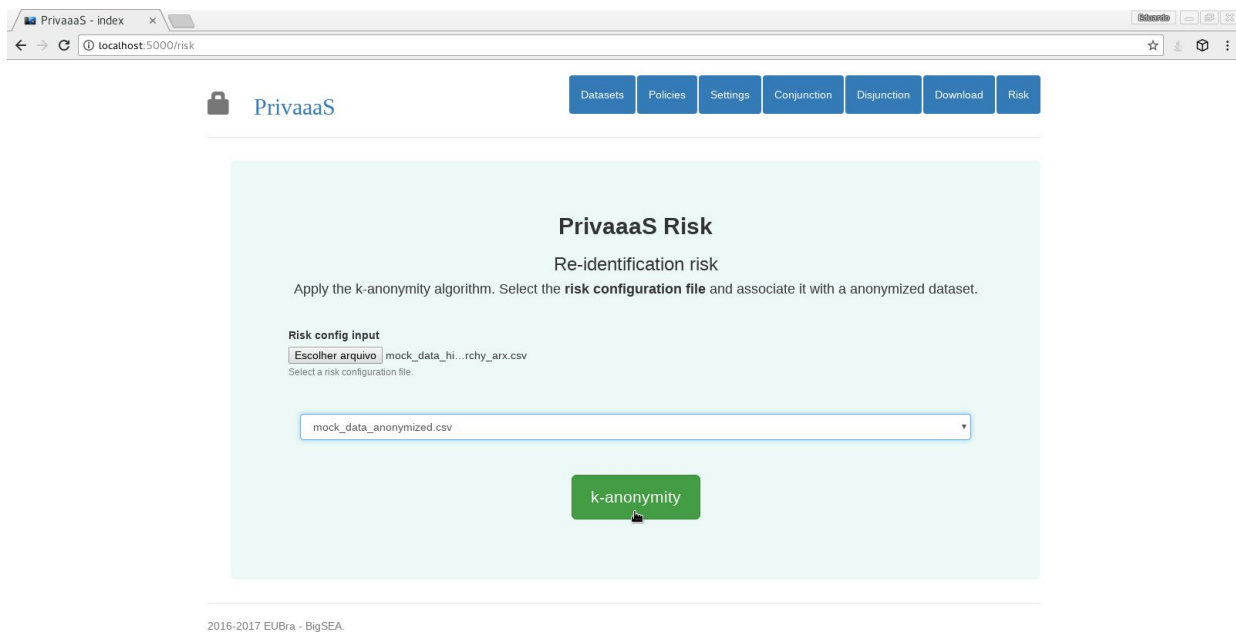
To apply k-anonymity, you must upload the risk configuration file accepted by the ARX anonymization library. Select the risk configuration file by clicking on "Select file".



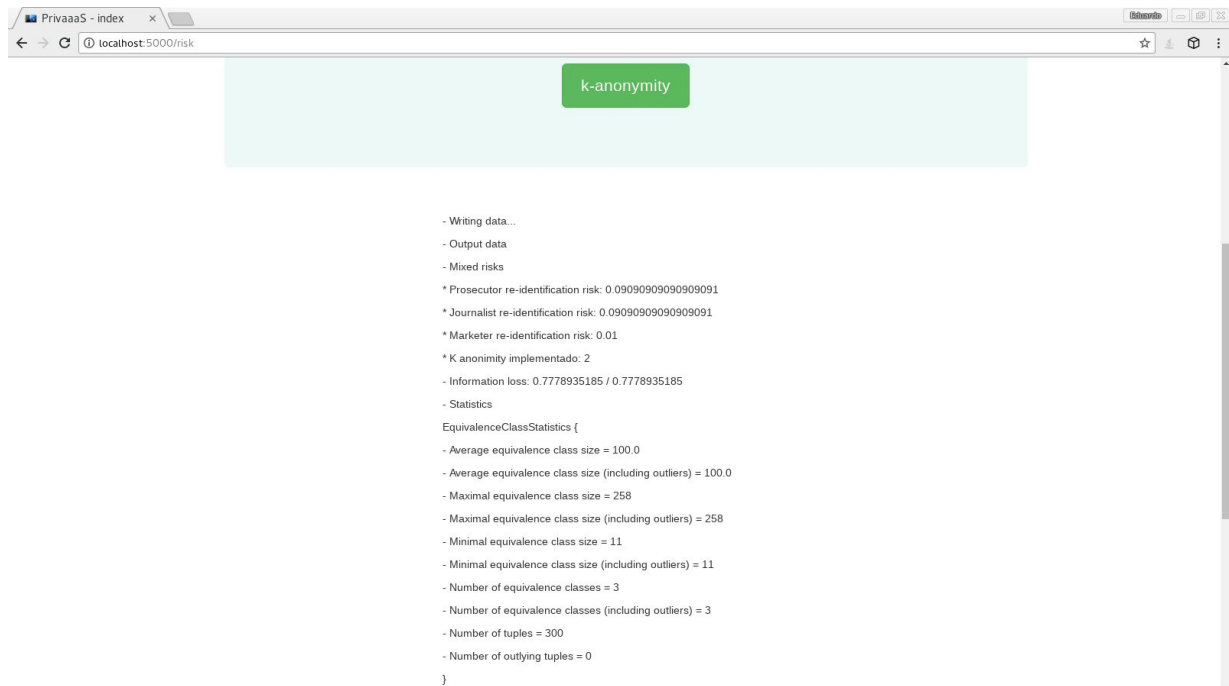
Then select one of the anonymized datasets corresponding to the risk configuration file.



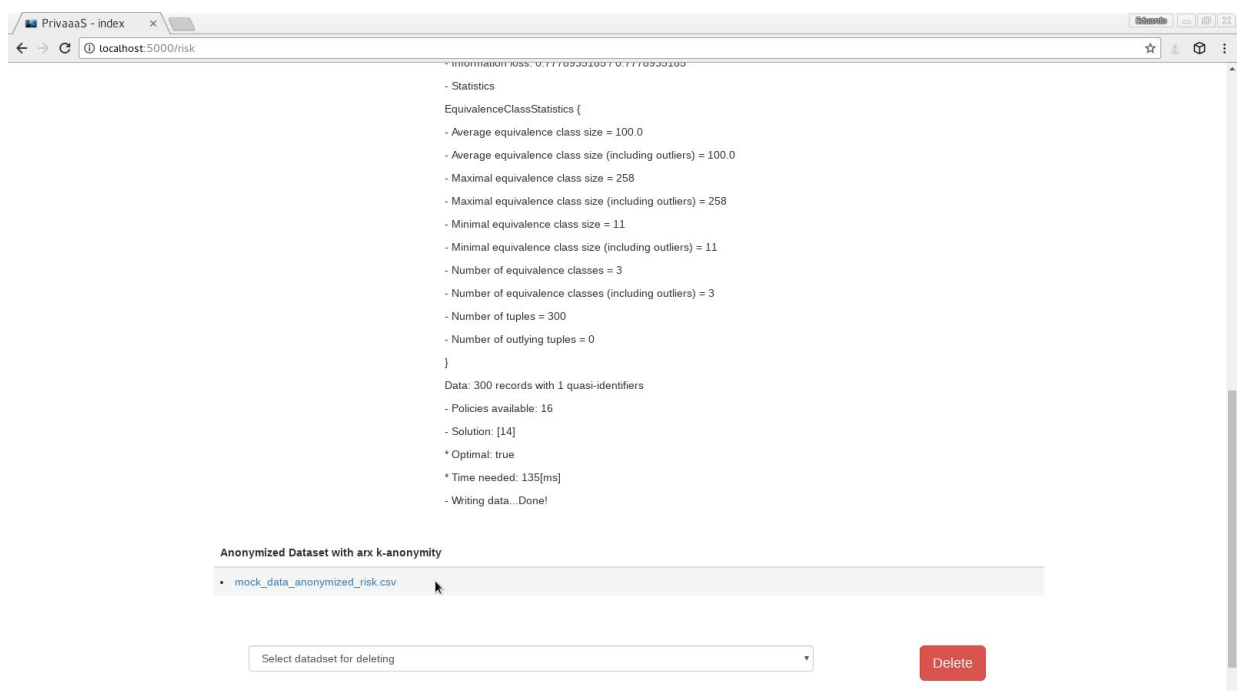
Click on "k-anonymity" to apply the algorithm.



ARX provides information on the risk of re-identification.

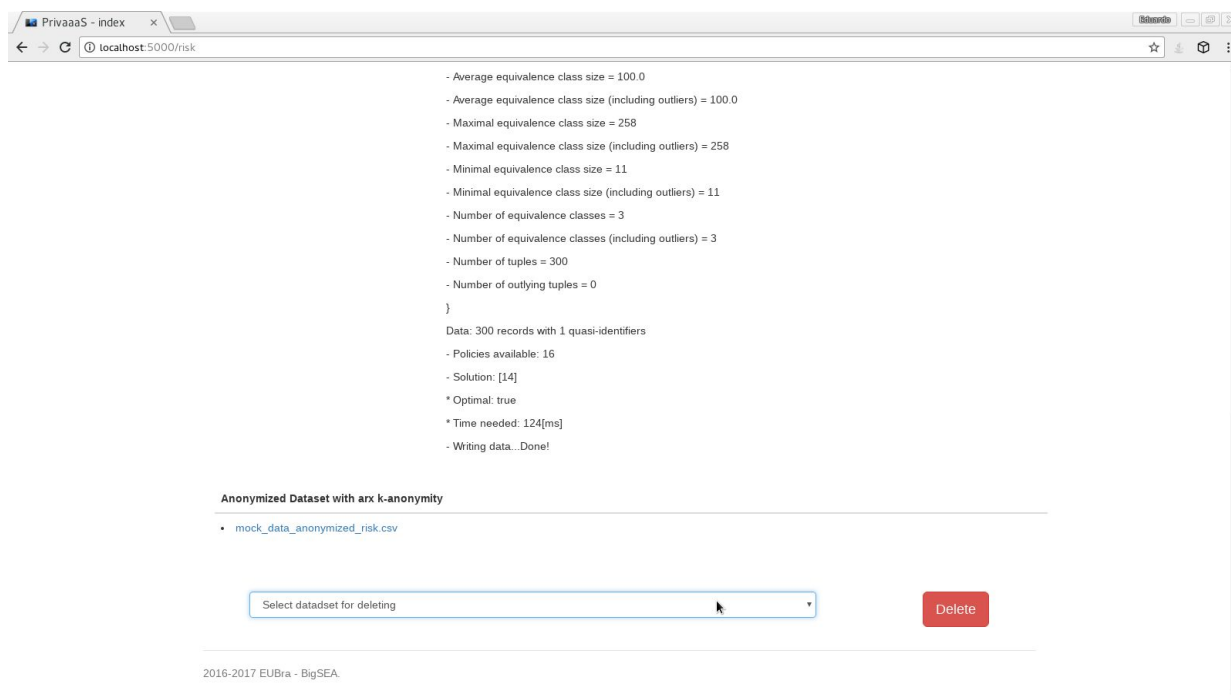


The output file, containing the anonymized dataset through the application of the k-anonymity algorithm will be available for download in the sequence.





You can delete the anonymized datasets in the same way as in the previous menus, i.e., selecting the file and clicking in the "Delete" button.



## cURL for command line - PrivaaaS

You can use PrivaaaS with cURL for the command line and use it with an available service in a URL. In this way, PrivaaaS can be used by other applications even if they were developed using different programming languages.

In the following instructions, DATASET\_NAME refers to the file name, PATH\_OF\_FILE refers to the file path and .EXT represents the type of file ( to this release are enabled json, csv and xml). POLICY\_NAME already refers to the name of the privacy policy and RISK\_NAME file refers to the filename that contains the risk configuration.

### 1. Sending datasets

To send datasets:

```
$ curl --form file_dataset=@PATH_OF_FILE/DATASET_NAME.EXT --form  
btn_send_dataset=Upload http://localhost:5000
```

or, if the file to be sent is in the same directory of the command:

```
$ curl --form file_dataset=@DATASET_NAME.EXT --form btn_send_dataset=Upload  
http://localhost:5000
```

Examples:

```
$ curl --form file_dataset=@greenData.csv --form btn_send_dataset=Upload  
http://localhost:5000
```

and

```
$ curl --form file_dataset=@mock_data.json --form btn_send_dataset=Upload  
http://localhost:5000
```

### 2. Deleting datasets

To delete a dataset:

```
$ curl --form del_list=DATASET_NAME.EXT --form delete=Delete http://localhost:5000
```

Example:

```
$ curl --form del_list=mock_data.json --form delete=Delete http://localhost:5000
```

### 3. Sending specific privacy policies

To send specific privacy policies:

```
$ curl --form file_policy=@POLICY_NAME.EXT --form datasets_list=DATASET_NAME.EXT  
--form btn_send_policy=Upload http://localhost:5000/policy
```

Examples:

```
$ curl --form file_policy=@dadosPessoais_policy.csv --form  
datasets_list=dadosPessoais.csv --form btn_send_policy=Upload  
http://localhost:5000/policy
```

and

```
$ curl --form file_policy=@mock_data_policy.json --form datasets_list=mock_data_json  
--form btn_send_policy=Upload http://localhost:5000/policy
```

### 4. Sending global privacy policies

To send global privacy policies:

```
$ curl --form file_policy=@GLOBAL_POLICY_NAME.EXT --form global_policy=True --form  
btn_send_policy=Upload http://localhost:5000/policy
```

Example:

```
$ curl --form file_policy=@AnonymizationStrategy_global_policies.xml --form  
global_policy=True --form btn_send_policy=Upload http://localhost:5000/policy
```

### 5. Deleting privacy policies

To delete a privacy policy:

```
$ curl --form del_list=POLICY_NAME.EXT --form delete=Delete http://localhost:5000/policy
```

Example:

```
$ curl --form del_list=mock_data_policy.json --form delete=Delete  
http://localhost:5000/policy
```

### 6. Generating anonymized datasets by conjunction process

To generate anonymized datasets by the Conjunction process:

```
$ curl --form btn_conjunction_send=Anonymize http://localhost:5000/privacy\_and
```

## 7. Generating anonymized datasets by disjunction process

To generate anonymized datasets by the Disjunction process:

```
$ curl --form btn_disjunction_send=Anonymize http://localhost:5000/privacy\_or
```

## 8. Downloading anonymized datasets

To download the anonymized datasets:

```
$ curl -O http://localhost:5000/static/up\_dir/output/ds\_files/DATASET\_NAME\_anonymized.json
```

or

```
$ curl -O http://localhost:5000/static/up\_dir/output/ds\_files/DATASET\_NAME\_anonymized.csv
```

## 9. Deleting anonymized datasets

To delete an anonymized dataset, depending on the file type:

```
$ curl --form del_list=DATASET_NAME_anonymized.json --form delete=Delete http://localhost:5000/privaaas
```

or

```
$ curl --form del_list=DATASET_NAME_anonymized.csv --form delete=Delete http://localhost:5000/privaaas
```

## 10. Calculating the re-identification risk and applying k-anonymity algorithm

To calculate the re-identification risk and apply the k-anonymity algorithm on the anonymized dataset:

```
$ curl --form file_hrisk=@RISK_NAME.csv --form select_anonymized_dataset=RISK_NAME_anonymized.csv --form btn_kanonymity_send=k-anonymity http://localhost:5000/risk
```

## 11. Downloading anonymized datasets by k-anonymity algorithm

To access and download the anonymized dataset by k-anonymity algorithm:

```
$ curl -O http://localhost:5000/static/up\_dir/output/h\_risk/DATASET\_NAME\_anonymized\_risk.csv
```

Example:

```
$ curl -O http://localhost:5000/static/up\_dir/output/h\_risk/dadosPessoais\_anonymized\_risk.csv
```

## 12. Deleting datasets anonymized using k-anonymity algorithm

To delete an anonymized file that used the k-anonymity:

```
$ curl --form del_list=DATASET_NAME_anonymized_risk.csv --form delete=Delete http://localhost:5000/risk
```

Example:

```
$ curl --form del_list=dadosPessoais_anonymized_risk.csv --form delete=Delete http://localhost:5000/risk
```

## 13. Example of use

Sending a json dataset:

```
$ curl --form file_dataset=@mock_data.json --form btn_send_dataset=Upload http://localhost:5000
```

Sending a csv dataset:

```
$ curl --form file_dataset=@personalData.csv --form btn_send_dataset=Upload http://localhost:5000
```

Sending a global privacy policy:

```
$ curl --form file_policy=@AnonymizationStrategy_global_policies.xml --form global_policy=True --form btn_send_policy=Upload http://localhost:5000/policy
```

Sending a specific privacy policy:

```
$ curl --form file_policy=@mock_data_policy.json --form datasets_list=mock_data_json --form btn_send_policy=Upload http://localhost:5000/policy
```

Setting conjunction process:

```
$ curl --form btn_conjunction_send=Anonymize http://localhost:5000/privacy\_and
```

Downloading the csv dataset anonymized:

```
$ curl -O http://localhost:5000/static/up\_dir/output/ds\_files/mock\_data\_anonymized.csv
```

Downloading the json dataset anonymized:

```
$ curl -O http://localhost:5000/static/up\_dir/output/ds\_files/personalData\_anonymized.json
```

Deleting all files:

```
$ curl --form del_list=mock_data.json --form delete=Delete http://localhost:5000
```

```
$ curl --form del_list=personalData.csv --form delete=Delete http://localhost:5000
```

```
$ curl --form del_list=AnonymizationStrategy_global_policies.xml --form delete=Delete http://localhost:5000/policy
```

```
$ curl --form del_list=mock_data_anonymized.csv --form delete=Delete http://localhost:5000/privaaas
```

```
$ curl --form del_list=mock_data_anonymized.json --form delete=Delete http://localhost:5000/privaaas
```

```
$ curl --form del_list=personalData_anonymized.csv --form delete=Delete http://localhost:5000/privaaas
```

```
$ curl --form del_list=personalData_anonymized.json --form delete=Delete http://localhost:5000/privaaas
```

## References

[1] Faker Provider (2017). [Online]. <https://faker.readthedocs.io/en/latest/providers.html>

[2] Faker Supported languages (2017). [Online]. <https://github.com/joke2k/faker>

[3] ARX. (2017) Arx data anonymization tool. [Online]. Available: <http://arx.deidentifier.org/>

[4] Basso, T., Moraes, R., Antunes, N., Vieira, M., Santos, W., & Meira Jr, W. (2017, May). PRIVAAaS: privacy approach for a distributed cloud-based data analytics platforms. In Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 1108-1116). IEEE Press.