Universidad de Vigo

bdpar:
Un paquete en R para ejecutar pipelines personalizables de preprocesados sobre fuentes de datos heterogéneos

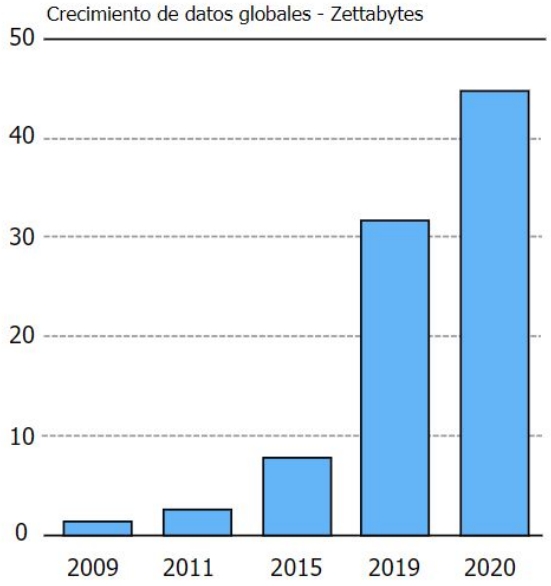
Autores: M. Ferreiro-Díaz, T. R. Cotos-Yañez, D. Ruano-Ordás
CRAN: https://cran.r-project.org/web/packages/bdpar/index.html
GitHub: https://github.com/miferreiro/bdpar

ÍNDICE

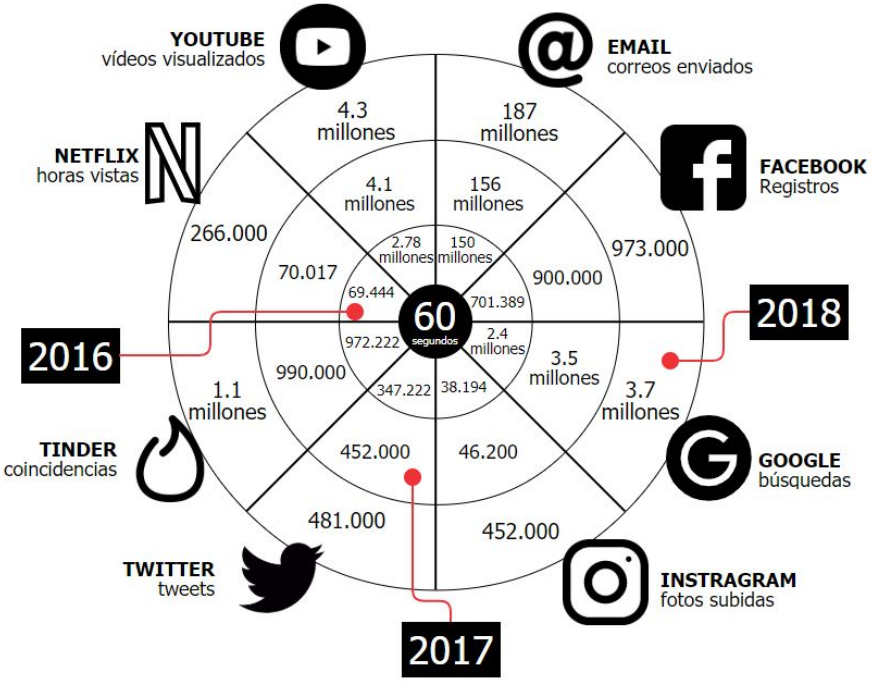
INTRODUCCIÓN

Crecimiento de los datos

Los datos crecen a una tasa anual compuesta del 40 por ciento, alcanzando cerca de 45 ZB para 2020

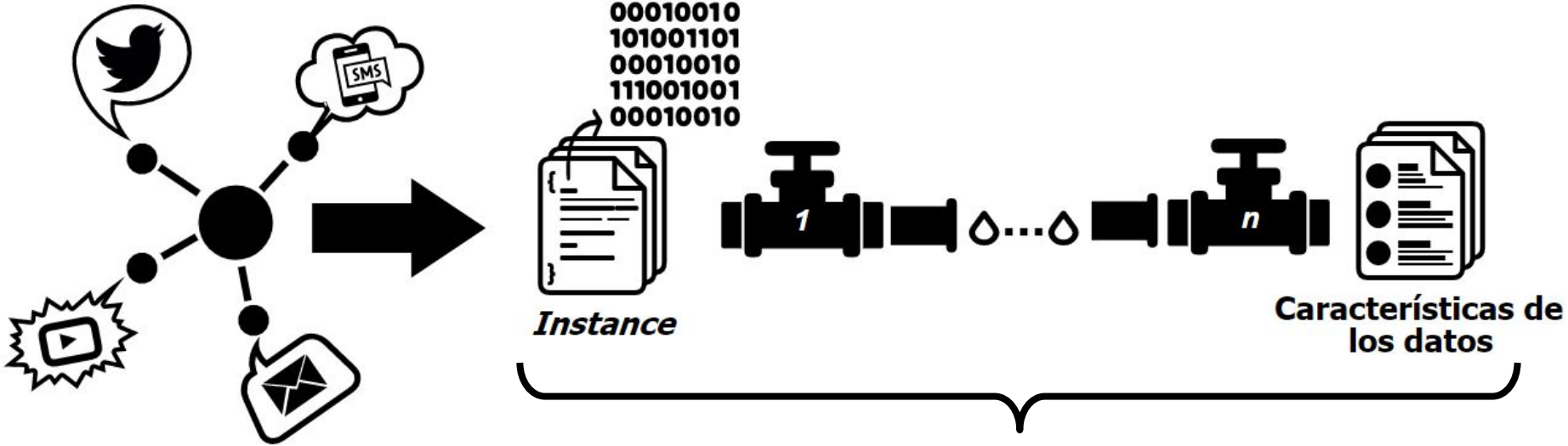


¿Qué pasa cada 60 segundos en Internet?



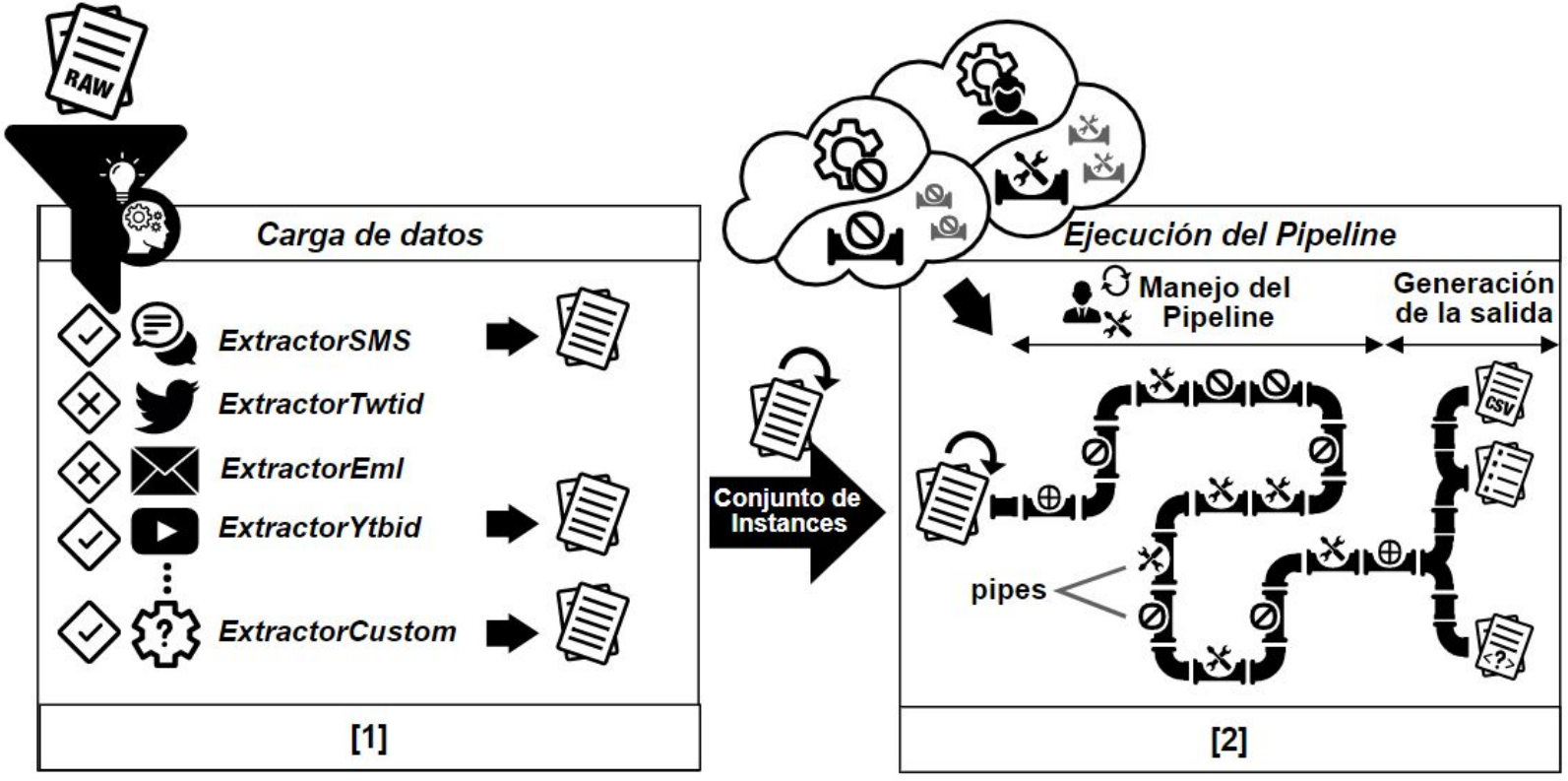
16%

OBJETIVOS

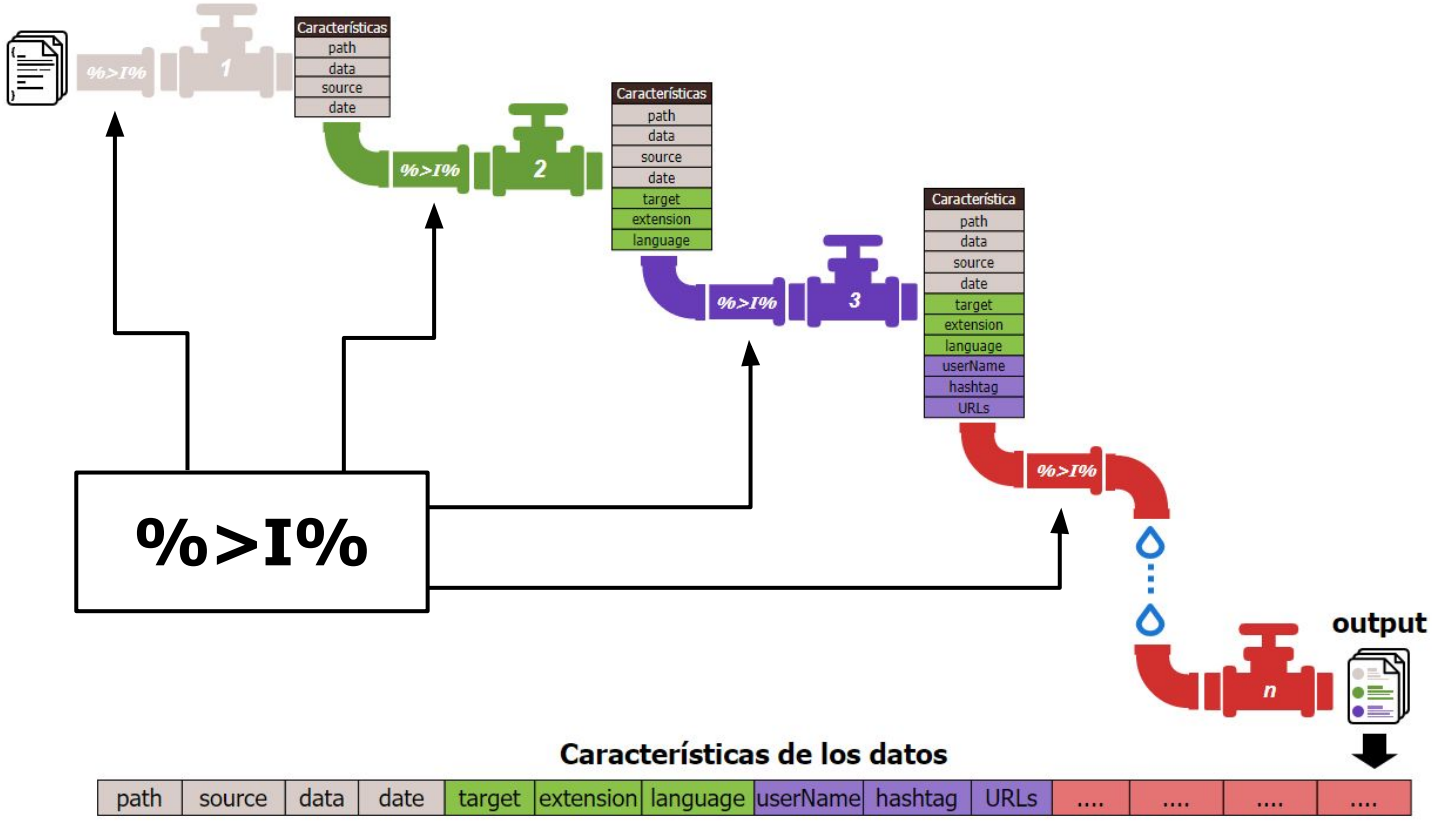


Implementación mediante clases R6

ARQUITECTURA



MODO DE FUNCIONAMIENTO (I)



MODO DE FUNCIONAMIENTO (II)

Pipes de funcionalidad básica

FindUserNamePipe

FindUrlPipe

FindHashtagPipe

FindEmojiPipe

FindEmoticonPipe

File2Pipe

GuessDatePipe

GuessLanguagePipe

TargetAssigningPipe

StoreFileExtensionPipe

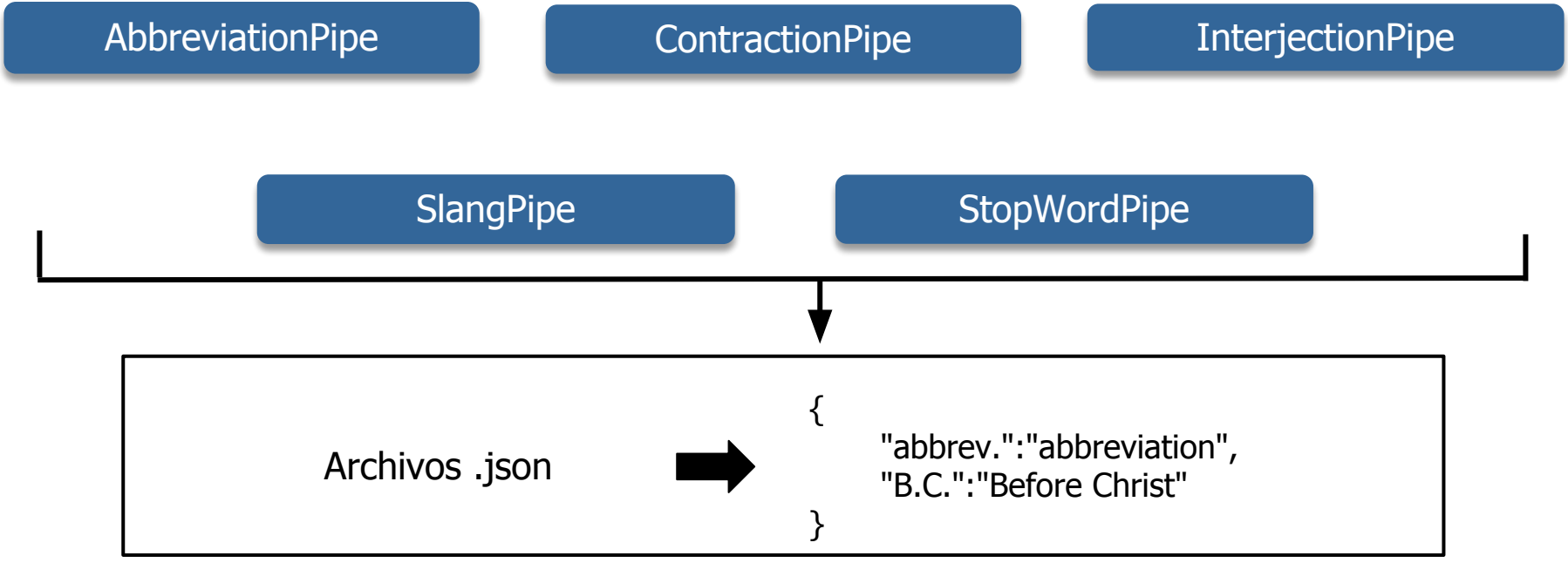
MeasureLengthPipe

ToLowerCasePipe

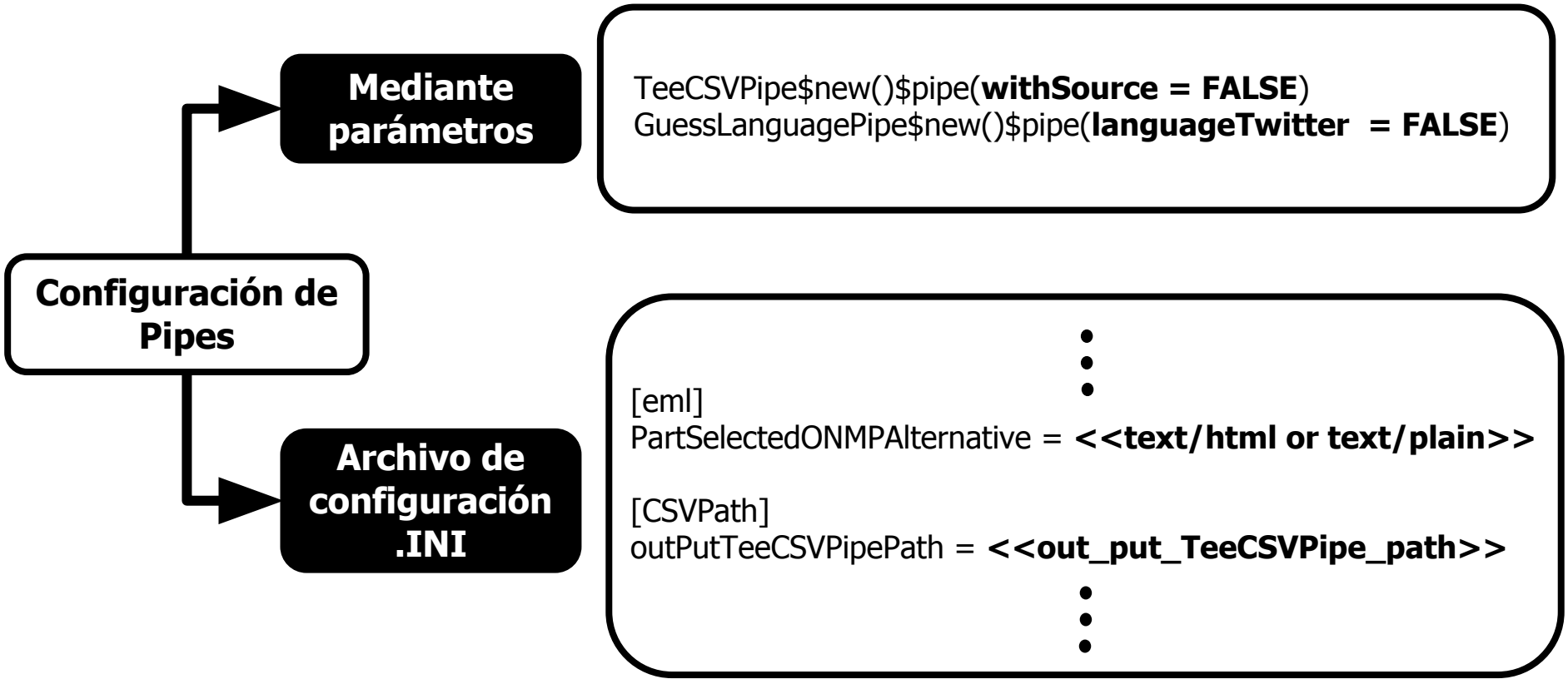
TeeCSVPipe

MODO DE FUNCIONAMIENTO (III)

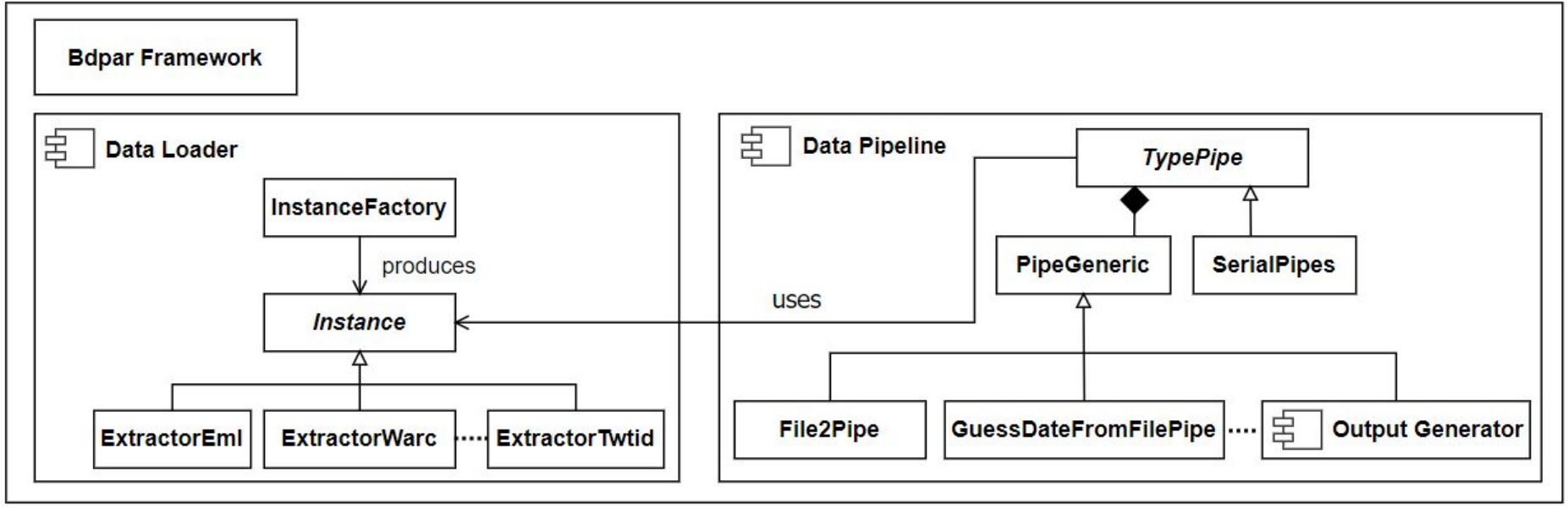
Pipes de acceso a ficheros externos



MODO DE FUNCIONAMIENTO (IV)



DISEÑO

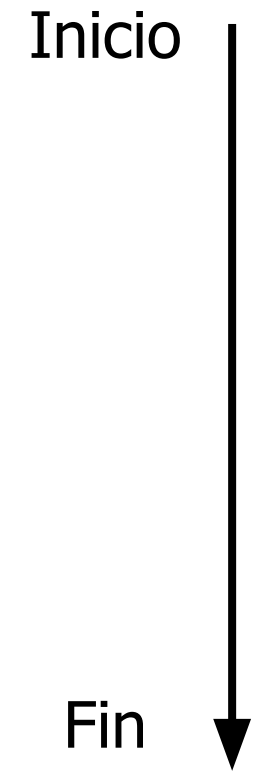

Bdpar Framework
Data Loader
InstanceFactory
produces
Instance
ExtractorEml
ExtractorWarc
ExtractorTwtid
Data Pipeline
TypePipe
PipeGeneric
SerialPipes
uses
File2Pipe
GuessDateFromFilePipe
Output Generator

Inicio del Pipeline

```
pipeline_execute(configurationFilePath = NULL,
                 editConfigurationFile = FALSE,
                 filesPath = "<folderWithFiles>",
                 pipe = SerialPipes$new(),
                 instanceFactory = InstanceFactory$new())
```

MANUAL DE USUARIO (II)

Flujo por defecto




```
instance %>I%
TargetAssigningPipe$new()$pipe() %>I%
StoreFileExtPipe$new()$pipe() %>I%
GuessDatePipe$new()$pipe() %>I%
File2Pipe$new()$pipe() %>I%
MeasureLengthPipe$new()$pipe("length_before_cleaning_text") %>I%
FindUserNamePipe$new()$pipe() %>I%
FindHashtagPipe$new()$pipe() %>I%
FindUrlPipe$new()$pipe() %>I%
FindEmoticonPipe$new()$pipe() %>I%
FindEmojiPipe$new()$pipe() %>I%
GuessLanguagePipe$new()$pipe() %>I%
ContractionPipe$new()$pipe() %>I%
AbbreviationPipe$new()$pipe() %>I%
SlangPipe$new()$pipe() %>I%
ToLowerCasePipe$new()$pipe() %>I%
InterjectionPipe$new()$pipe() %>I%
StopWordPipe$new()$pipe() %>I%
MeasureLengthPipe$new()$pipe("length_after_cleaning_text") %>I%
TeeCSVPipe$new()$pipe()
```

MANUAL DE USUARIO (III)

Nuevo flujo de preprocesamiento

```
01  pipe = function(instance) {
02    if(!"Instance" %in% class(instance)) {
03      stop("[RemovesWhiteSpaces][pipe][Error]",
04           Checking the type of the variable: instance ",
05             class(instance))
06    }
07    instance$addFlowPipes("RemovesWhiteSpaces")
08    if (!instance$checkCompatibility("RemovesWhiteSpaces", self$getAlwaysBeforeDeps())){
09      stop("[RemovesWhiteSpaces][pipe][Error] Bad compatibility between Pipes.")
10    }
11    instance$addBanPipes(unlist(super$getNotAfterDeps()))
12
13    instance$getData() %>>%
14      stringr::str_trim() %>>%
15        stringr::str_squish() %>>%
16          instance$setData()
17
18    return(instance)
19  }
```

RemovesWhiteSpaces.R

```
01  pipeAll = function(instance) {
02    if(!"Instance" %in% class(instance)) {
03      stop("[RemovesWhiteSpaces][pipe][Error]",
04           Checking the type of the variable: instance ",
05             class(instance))
06    }
07    message("[TestPipe][pipeAll][Info]", instance$getPath(), "\n")
08    tryCatch(
09      instance %>>I%
10        TargetAssigningPipe$new()$pipe() %>>I%
11        StoreFileExtensionPipe$new() %>>I%
12        File2Pipe$new()$pipe() %>>I%
13        RemovesWhiteSpaces$new()$pipe() %>>I%
14        TeeCSVPipe$new()$pipe()
15    ,
16    error = function(e) {
17      message("[TestPipe][pipeAll][Error]", instance$getPath(), ":", paste(e), "\n")
18      instance$invalidate()
19    })
20    return(instance)
21  }
```

TestPipe.R

```
01  library(bdpar);library(R6);library(pipeR);library(stringr)
02  source("TestPipe.R");source("RemovesWhiteSpaces.R")
03
04  output <- pipeline_execute(configurationFilePath = "configurations.ini",
05                             editConfigurationFile = FALSE,
06                             filesPath = "testFiles",
07                             pipe = TestPipe$new(),
08                             instanceFactory = InstanceFactory$new())
```

main.R

MANUAL DE USUARIO (III)

Nuevo flujo de preprocesamiento

Validación

Funcionalidad

```
01  pipe = function(instance) {
02    if(!"Instance" %in% class(instance)) {
03      stop("[RemovesWhiteSpaces][pipe][Error]",
04           Checking the type of the variable: instance ",
05             class(instance))
06    }
07    instance$addFlowPipes("RemovesWhiteSpaces")
08    if (!instance$checkCompatibility("RemovesWhiteSpaces", self$getAlwaysBeforeDeps())){
09      stop("[RemovesWhiteSpaces][pipe][Error] Bad compatibility between Pipes.")
10    }
11    instance$addBanPipes(unlist(super$getNotAfterDeps()))
12
13    instance$getData() %>>%
14      stringr::str_trim() %>>%
15        stringr::str_squish() %>>%
16          instance$setData()
17
18    return(instance)
19  }
```

RemovesWhiteSpaces.R

MANUAL DE USUARIO (III)

Nuevo flujo de preprocesamiento

Validación

Funcionalidad

```
01  pipeAll = function(instance) {
02    if(!"Instance" %in% class(instance)) {
03      stop("[RemovesWhiteSpaces][pipe][Error]",
04           Checking the type of the variable: instance ",
05             class(instance))
06    }
07    message("[TestPipe][pipeAll][Info]", instance$getPath(), "\n")
08    tryCatch(
09      instance %>I%
10        TargetAssigningPipe$new()$pipe() %>I%
11        StoreFileExtensionPipe$new() %>I%
12        File2Pipe$new()$pipe() %>I%
13        RemovesWhiteSpaces$new()$pipe() %>I%
14        TeeCSVPipe$new()$pipe()
15    ,
16    error = function(e) {
17       message("[TestPipe][pipeAll][Error]", instance$getPath(), ":", paste(e), "\n")
18       instance$invalidate()
19    })
20    return(instance)
21  }
```

TestPipe.R

MANUAL DE USUARIO (III)

Nuevo flujo de preprocesamiento

```
01  library(bdpar);library(R6);library(pipeR);library(stringr)
02  source("TestPipe.R");source("RemovesWhiteSpaces.R")
03
04  output <- pipeline_execute(configurationFilePath = "configurations.ini",
05                             editConfigurationFile = FALSE,
06                             filesPath = "testFiles",
07                             pipe = TestPipe$new(),
08                             instanceFactory = InstanceFactory$new())
```

main.R

EJEMPLO(I)

Planteamiento

- **Datos de prueba**
  - **Temática: alimentación**
  - **Descripción de archivos:**
    - **10 SMS spam**
    - **10 SMS no spam**
    - **10 emails spam**
    - **10 emails no spam**
- **Flujo de preprocesamiento utilizado** → **SerialPipes**
- **Archivo de configuración .INI**

```
[resourcesPath]
resourcesAbbreviationsPath = testResources/abbreviations-json
resourcesContractionsPath = testResources/contractions-json
resourcesInterjectionsPath = testResources/interjections-json
pathResourcesSlangsPath = testResources/slangs-json
resourcesStopWordsPath = testResources/stopwords-json

[eml]
PartSelectedOnMPAlternative = text/plain

[CSVPath]
outPutTeeCSVPipePath = example.csv
```

EJEMPLO(II)

Lanzamiento del preprocesamiento

pipeline_execute(**configurationFilePath** = "configurations.ini",
filesPath = "testFiles")

Características obtenidas

path	target	userName	contractions
data	extension	hashtag	abbreviation
source	length_before_cleaning	URLs	lang_prop_name
date	length_after_cleaning_text	emoticon	interjection
Initial_path	language	Emojis	stopWord

EJEMPLO(III)

Wordclouds obtenidos

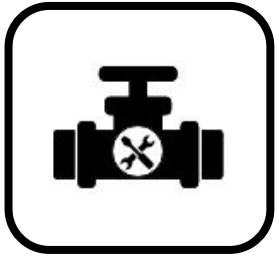
SMS

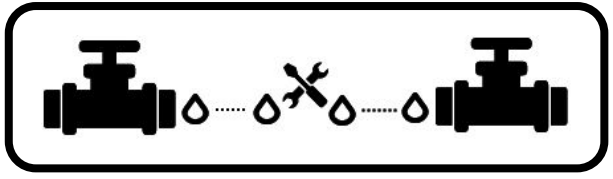
EML

SMS + EML

CONCLUSIONES

Personalización de funcionalidades

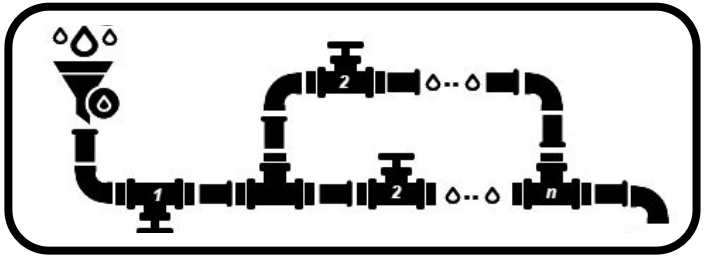




Nuevo operador

%>I%

Futuras ampliaciones



REFERENCIAS

- Paquete bdpar:
  - CRAN: https://cran.r-project.org/web/packages/bdpar/index.html
  - GitHub: https://github.com/miferreiro/bdpar
- Aplicación en clasificación:
  - GitHub: https://github.com/miferreiro/clasification
- Librerías utilizadas:

Imports

ini	magrittr	pipeR
purrr	R6	rlist
svMisc	tools	utils

Suggests

cld2	knitr	readr
rex	rjson	rmarkdown
rtweet	stringi	stringr
textutils	tuber	

Universidade Vigo

bdpar:
Un paquete en R para ejecutar pipelines personalizables de preprocesados sobre fuentes de datos heterogéneos

Autores: M. Ferreiro-Díaz, T. R. Cotos-Yañez, D. Ruano-Ordás
CRAN: https://cran.r-project.org/web/packages/bdpar/index.html
GitHub: https://github.com/miferreiro/bdpar