



# Python ANTLR4 Demo

CARLOS LORIA-SAENZ

II-2018. PARA USO EXCLUSIVO DEL CURSO

- ▶ Este material no debe ser redistribuido sin consultar al autor.

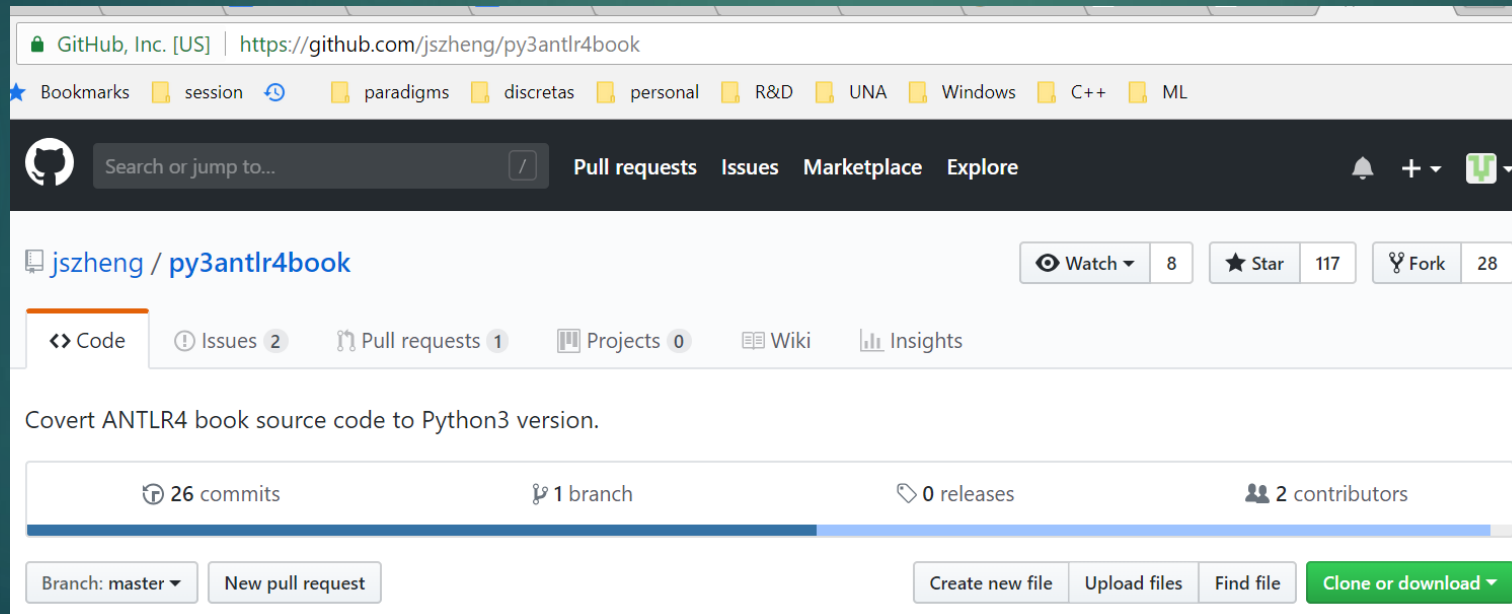
# Introducción

- ▶ Un demo que parsea e imprime expresiones de lógica proposicional en notación Wang
- ▶ Incluye instalación y uso

# Referencia Principal

4

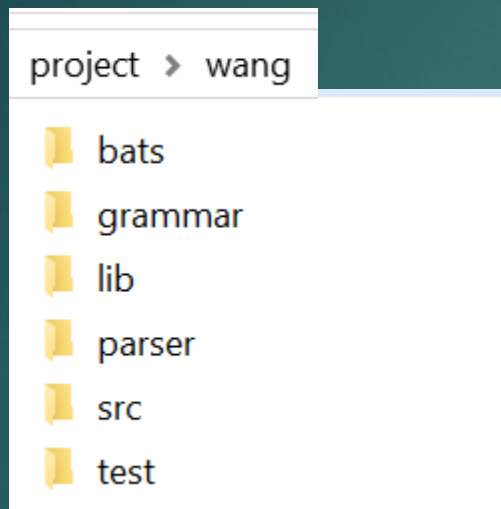
- ▶ La documentación del runtime de ANTLR4 para Python3 está disponible [acá](#), (autor: jszheng) se adaptó y extendió ejemplo `calc` para que este material funcione



- ▶ Generador de parsers usando técnicas top-down (LL)
- ▶ Genera árbol de parsing y visitadores para ir “bajando” por el árbol (patrón Visitor)
- ▶ Está en Java.
- ▶ Por defecto genera Java
- ▶ Tiene otros puertos (C#, Python JS)
- ▶ Aquí queremos usarlo en Python por facilidad con el proyecto programado

# El poyecto inicial recibido

6



# Instalación un ambiente virtual

7

Este paso es innecesario, si se instaló el runtime de antlr4 python3 de manera global

Se asume se instala en project

```
PL:virtualenv EIF400_ENV
Using base prefix 'c:\\python36'
New python executable in D:\\UNA\\Paradigmas\\CursoLenguajesDePro
ripts\\python.exe
Installing setuptools, pip, wheel...done.
```

PL:

EIF400\_ENV

wang

- bats
- grammar
- lib
- parser
- src
- test

# Instalación runtime antlr4-python

8

```
PL:EIF400_ENV\Scripts\activate
```

```
(EIF400_ENV) PL:pip install antlr4-python3-runtime
```

```
Collecting antlr4-python3-runtime
```

```
Using cached https://files.pythonhosted.org/packages/3e/96/aba01b2948ec67f2e/antlr4-python3-runtime-4.7.1.tar.gz
```

```
Building wheels for collected packages: antlr4-python3-runtime
```

```
Running setup.py bdist_wheel for antlr4-python3-runtime ... done
```

```
Stored in directory: C:\Users\una\AppData\Local\pip\Cache\wheels\ef\f6\18\aeacb10
```

```
Successfully built antlr4-python3-runtime
```

```
Installing collected packages: antlr4-python3-runtime
```

```
Successfully installed antlr4-python3-runtime-4.7.1
```

```
(EIF400_ENV) PL:
```

```
(EIF400_ENV) PL:pip list
```

Package	Version
-----	-----
antlr4-python3-runtime	4.7.1
pip	18.0
setuptools	40.4.1
wheel	0.31.1

```
(EIF400_ENV) PL:
```



# Runtime de ANTLR4 4 (la última versión es 4.7.1 al hacer este ppt)

► [Visite Download](#)

## ANTLR tool and Java Target

- Complete ANTLR 4.7.1 Java binaries jar. Complete ANTLR 4.7.1 tool, Java runtime and ST 4.0.8, which lets you run the tool and the generated code.

<http://www.antlr.org/download/antlr-4.7.1-complete.jar>

# Copie antlr jar

10

- ▶ Por simpleza pondremos el JAR en C:\antlr\lib
- ▶ Si usa otro lugar edite bats\antlr4env.bat según corresponda
- ▶ Ejecute ese bat en el directorio wang

\\bats\antlr4env.bat

```
18/09/2018 05:58 <DIR> .
18/09/2018 05:58 <DIR> ..
09/09/2016 06:17 <DIR> grammar
25/08/2018 04:23 <DIR> parser
25/08/2018 04:41 <DIR> lib
25/08/2018 04:42 <DIR> test
25/08/2018 04:43 <DIR> src
25/08/2018 04:45 <DIR> bats
0 archivos 0 bytes
8 dirs 14 485 962 752 bytes libres
```

```
PL:bats\antlr4env.bat
*** Setting Environment and Variables for ANTLR4-Python ***
VARIABLES
*** grammar_path=.\grammar\Wang.g4 ***
*** parser_path=.\parser ***
*** classpath=C:\antlr\lib\antlr-4.7.1-complete.jar;. ***
*** pythonpath=.\parser\grammar;.src; ***
COMMANDS
*** Use 'antlr4' for calling directly to antlr4
*** Use 'antlr4p3' for generating parsers for python3 target ***
*** Use 'build_parser' for generating parser from .\grammar\Wang.g4 into .\parser ***
PL:
```

```
8 REM loriacarlos@gmail.com
9 echo *** Setting Environment and Variables for ANTLR4-Python ***
10 SET ANTLR_JAR=C:\antlr\lib\antlr-4.7.1-complete.jar
```

# Pruebe que el JAR es visible así:

11

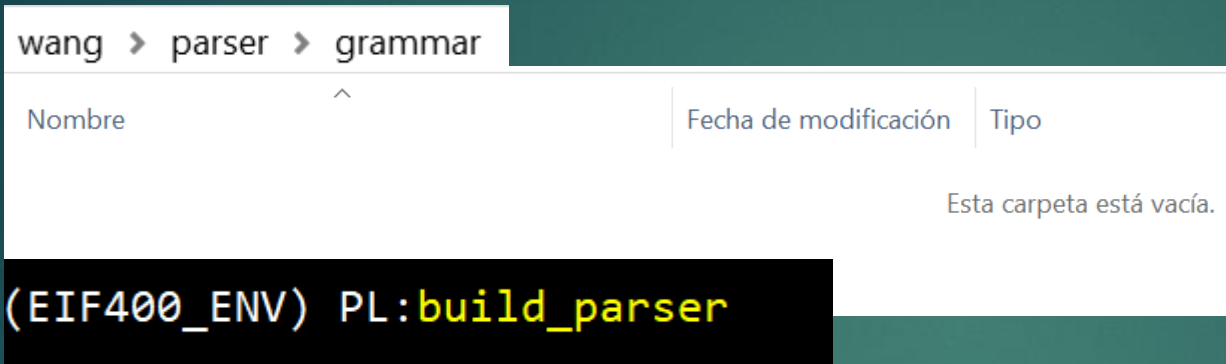
```
PL:antlr4
```

```
ANTLR Parser Generator Version 4.7.1
```

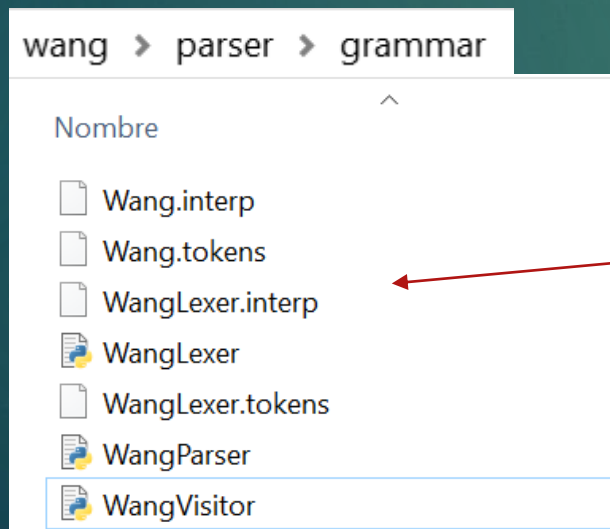
```
-o ____          specify output directory where all output is generated
-lib ____        specify location of grammars, tokens files
-atn             generate rule augmented transition network diagrams
-encoding ____   specify grammar file encoding; e.g., euc-jp
-message-format ____ specify output style for messages in antlr, gnu, vs2005
-long-messages   show exception details when available for errors and warnings
-listener        generate parse tree listener (default)
-no-listener     don't generate parse tree listener
-visitor         generate parse tree visitor
-no-visitor      don't generate parse tree visitor (default)
-package ____    specify a package/namespace for the generated code
-depend          generate file dependencies
-D<option>=value set/override a grammar-level option
-Werror          treat warnings as errors
-XdbgST          launch StringTemplate visualizer on generated code
-XdbgSTWait      wait for STViz to close before continuing
-Xforce-atn      use the ATN simulator for all predictions
-Xlog            dump lots of logging info to antlr-timestamp.log
-Xexact-output-dir all output goes into -o dir regardless of paths/package
```

# Generar parser

12



Carpeta vacía antes de generar parser



Parser generado correctamente

# Para correr: use script wang.py

13

```
(EIF400_ENV) PL:python src\wang.py test\t0.wang
*** Testing Wang Parser (EIF400 II-2018) ***
*** Processing from file "test\t0.wang" ***
Data:
(p | r) & ~q => q.
p=>q.
=> q.
*** Starts visit of data ***

Start Visiting FormExpr (=>) 2 children
Visiting SeqExpr (,) with 1 children
Visiting AndExpr (&)
Visiting ParenExpr (...)
Visiting OrExpr (|)
Visiting Id=p
Visiting Id=r
Visiting NotExpr (~)
Visiting Id=q
Visiting SeqExpr (,) with 1 children
Visiting Id=q
```

Salida fue truncada

# Detalles

14

- ▶ El resto será explicado más en detalle en clase