**DSL orientado a Scrum**

Este proyecto consiste en el desarrollo de un DSL (Domain-Specific Language) orientado a Scrum, con una interpretación a Python. El objetivo principal del DSL es facilitar la gestión de roles, historias de usuario y otras actividades relacionadas con el desarrollo ágil de software. El DSL ofrece dos modalidades principales para la creación y gestión de roles e historias de usuario.

**Modo Plantilla:**

En este modo, el DSL proporciona una plantilla prediseñada para la definición de roles y la creación de historias de usuario. El usuario solo necesita asignar los roles a los miembros del equipo y crear las historias de usuario. La asignación de roles a las historias de usuario se realiza automáticamente de forma aleatoria.

**Modo Creativo:**

En este modo, el usuario tiene la libertad de definir funciones personalizadas para la asignación de roles a las historias de usuario. Puede crear funciones que asignen los roles de forma aleatoria o según criterios específicos definidos por el usuario. Este modo ofrece una mayor flexibilidad y control sobre la asignación de roles.

BNF

// Tipos básicos

string ::= "\" characters "\"

characters ::= alphabet\*

int ::= [0-9]+

alphabet ::= [a-zA-Z]+

Identifier ::= /[a-zA-Z\_$][0-9a-zA-Z\_$]\*/

// Definición de Roles y Sprints

roles ::= ”RS” “{” role ("," role)\* “}”

role ::= role\_type ":" role\_name

role\_name ::= string

role\_type ::= scrumMaster | productOwner | developer | tester

scrumMaster ::= “SM”

productOwner ::= “PO”

developer ::= “DV”

tester ::= “QA”

// Historias de Usuario

userStories ::= userStory+

userStory ::= "US" userStoryID "{"

UserStoryFormatBlock

"}"

UserStoryFormatBlock ::= "T" ":" title ","

"TY" ":" userStoryType ","

"PS" ":" assignedTo ","

("QA" ":" tester ",")?

"DS" ":" description ","

"ET" ":" estimation ","

"AC" ": acceptance

title ::= string

userStoryType ::= “Feature” | “Spike” | “POC” | “Fix” | “HotFix”

assignedTo ::= role

description ::= string

estimation ::= int

acceptance ::= string

userStoryID ::= string

method ::= “funct” “{“ statment “}”

statement ::= conditional | loop

acceptedData ::= role\_type | estimation | userStoryType | int

// Condicionales

conditional ::= "if" "(" condition ")" "{" block "}" ("else" "{" block "}")?

condition ::= expression

expression ::= condition\_expression

condition\_expression ::= "True" | "False" | comparison

comparison ::= acceptedData comparator acceptedData

comparator ::= "==" | "!=" | "<" | ">" | "<=" | ">="

// Bucles

loop ::= "for" "(" acceptedData"in" iterable ")" "{" block "}"

iterable ::= "Statuses" | "Roles" | "Sprints" | "UserStories" | "Ceremonies"

// Bloque de código

block ::= statement\*

statement ::= roles | userStories | conditional | loop | method

randomUser ::= “rm”

**Ejemplos**

<https://drive.google.com/file/d/16HG0VpVMiNAbZRRmRbdcuTEzwXmvwmOn/view?usp=sharing>

Eg

string ::= "\" characters "\""`

```

"HelloWorld"

```

Identifier ::= /[a-zA-Z\_$][0-9a-zA-Z\_$]\*/

```

myVariable1

```

role ::= role\_name ":" role\_type

```

Samuel:DV

```

roles ::= ”RS” “{” role ("," role)\* “}”

```

roles {

SM : "Alice" ,

PO : "Bob",

DV : "Charlie" ,

DV: "Dave",

QA : "Eve"

}

```

userStory ::= "US" userStoryID "{"

UserStoryFormatBlock

"}"

```

"US-1" {

T:"Implement login",

TY:"Feature",

PS:"Diego",

DS:"Create login functionality",

ET:3,

AC:"Login works successfully"

}

```

loop ::= "for" "(" acceptedData"in" iterable ")" "{" block "}"

```

for (DV in Roles) {

block

}

```

statement ::= roles | userStories | conditional | loop | method

```

for (DV in Roles) { userStories }

```

```

roles {

SM : "Alice" ,

PO : "Bob",

DV : "Charlie" ,

DV: "Dave",

QA : "Eve"

}

// Definición de User Stories

userStories {

US "US1" {

T: "Implement login feature",

TY: Feature,

PS: DV : "Charlie"

DS: "As a user, I want to log in to access my account.",

ET: 5,

AC: "User can log in successfully",

UST: "Design", "Development", "Testing"

},

US "US2" {

T: "Fix login bug",

TY: Fix,

PS: QA : "Eve",

DS: "Fix the bug in the login feature.",

ET: 3,

AC: "User can log in without errors",

UST: "Debugging", "Testing"

}

}

// Método personalizado para asignación aleatoria

funct assignRandom {

for (userStory in UserStories) {

if (userStory.PS == "") {

userStory.PS = roles[randomInt(0, len(roles)-1)]

}

}

}

```