

DESCUBRIENDO PERL6

ANTONIO GÁMIZ

CAMELIA, EL BICHO DE PERL6





Perl 6 unifies many great ideas that aren't usually found in other programming languages.

MULTIPARADIGMA

OOP

```
class Perl6::OOP {  
    has Str %atributos;  
  
    method say() { ... }  
}  
  
role Pintar {  
    (...)  
}
```

FUNCIONAL

```
# Para ver cómo es esto pásate por la charla de JJ  
# a las 15:00!
```

UNOS CUANTICOS MÁS

```
# Declarativa  
# Imperativa  
# Y otros tantos que no entiendo
```

UNICODE!

```
say "\c[family: man woman girl boy]"; # OUTPUT: «𠁿𠁿𠁿𠁿»
```

```
my $var = ¼₀ + 2 + XII;
say $var;                      # OUTPUT: «14.1»
```

```
A ∪ U === U;
A ∩ U === A;
A ∩ Ø === Ø;
A ∪ Ø === A;
my @a = <1 2 3>;
my @b = <4 5 6>;
say (@a, @b) »[1]; # OUTPUT: «(2 5)»
say (@a, @b) >> [1]; # OUTPUT: «(2 5)»
```

HAZLO COMO QUIERAS

```
# Tipos
my $sth;                                # lo que quieras
my Str @text = ["array", "string"]; # Array de Str
my Int %hash{Str} = a => 1, b => 2; # hash de Str a Int
```

```
# Incluso subtipos
subset File of Str where .IO.e      # archivos existentes

sub read-file(File $f) {
    ...
}
```

```
# Argumentos
sub a ($whatever) { ... }
sub b (:$named)   { ... }

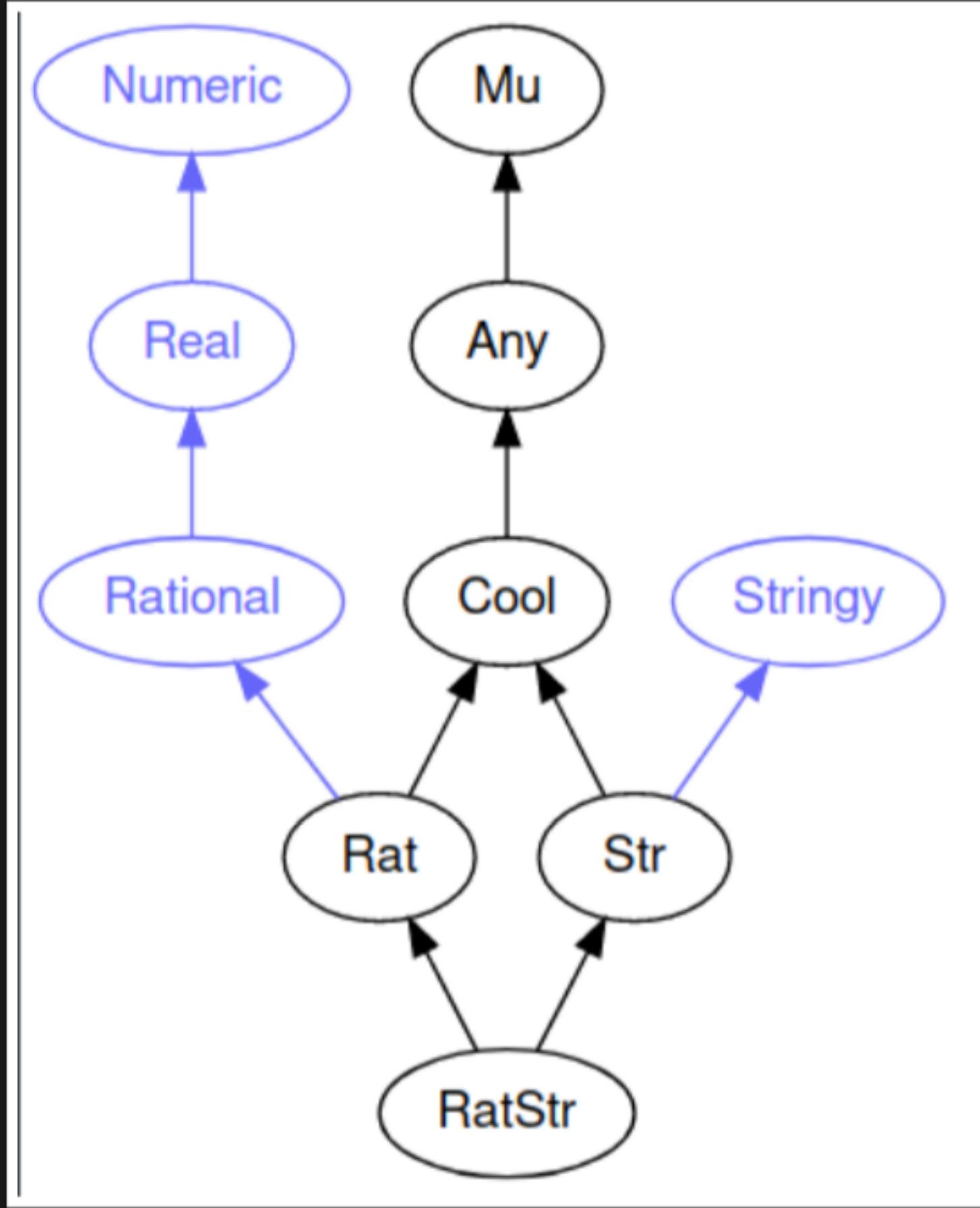
# Multi rutinas
multi sub stringify(0) { say "zero" }
multi sub stringify($) { say "lo que sea" }
```

GRAMÁTICAS!

```
unit grammar Perl6::TypeGraph::Decl;

token ident      { <.alpha> \w* }
token apostrophe { <[ ' \-' ]> }
token identifier { <.ident> [ <.apostrophe> <.ident> ]* }
token longname   { <identifier>+ % '::'
token package    { class | module | role | enum }
token rolesig     { '[' <-[ \[\] ]>* ']' }
rule inherits    { 'is' <longname> }
rule roles        { 'does' <longname><rolesig>?

rule TOP {
    ^
    <package>
    <longname><rolesig>?
```



```
[Basic]
# Base types
class Mu
class Junction
class Any
class Scalar
```

is Mu
is Mu
is Any

SISTEMA DE MÓDULOS: ZEF

[HTTPS://MODULES.PERL6.ORG/](https://modules.perl6.org/)

DOCUMENTACIÓN!

[HTTPS://DOCS.PERL6.ORG/](https://docs.perl6.org/)

POD 6

```
=begin pod

=head1 Los objetos también tienen documentación!

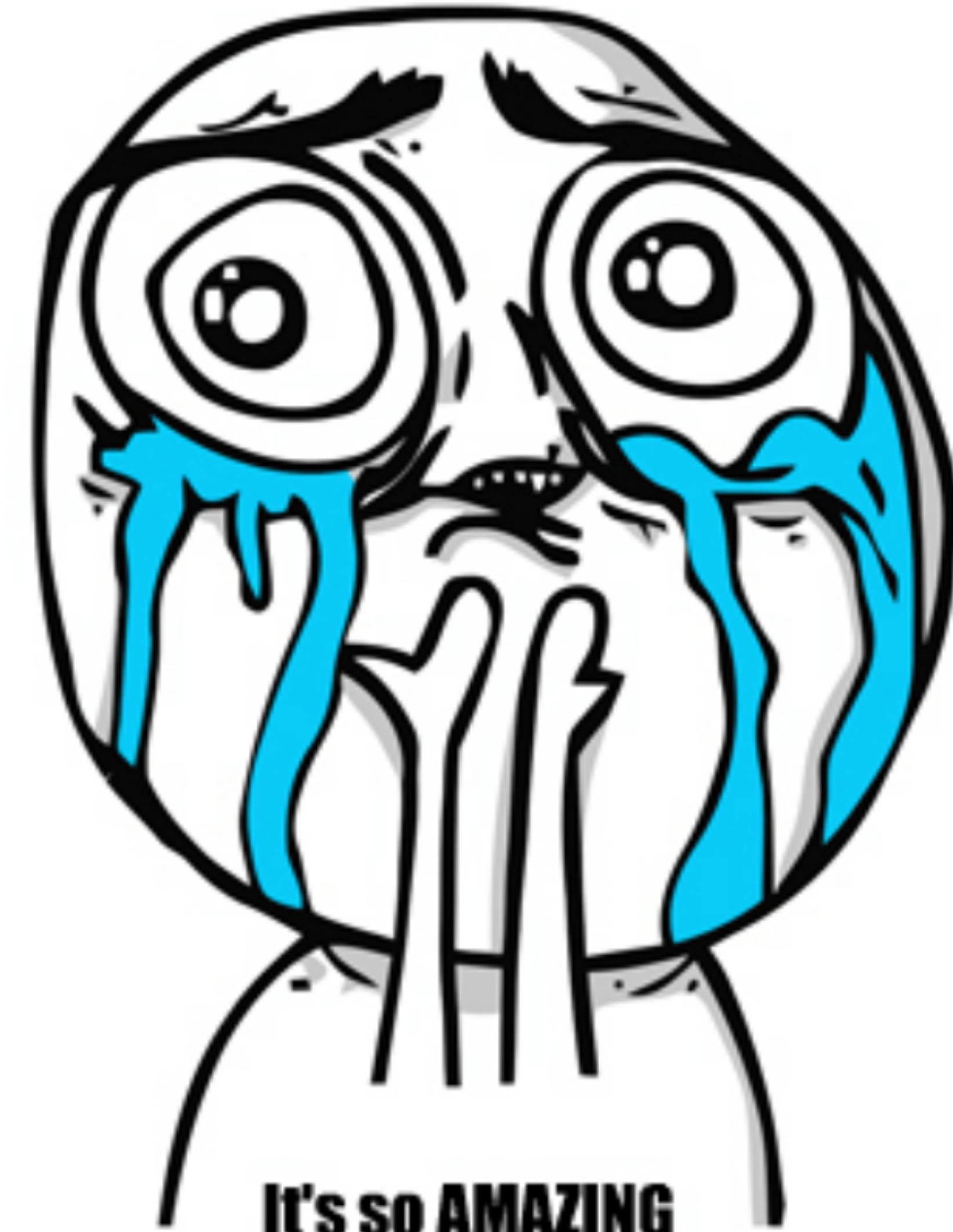
En Perl6 todo es un objeto.

=begin code
# | Este comentario es parte de la función!
sub alguna-funcion() { . . . }
=end code

=end pod
```

¿QUIERES APRENDERLO?

- <https://docs.perl6.org/>
- Perl6 Weekly Challenge
- IRC Channel en freenode => #perl6
- Advent calendar



It's so AMAZING

memecenter.com

