Perldoop 2.0

Un compilador fuente-a-fuente Perl-Java

Grado en Ingeniería Informática
Universidad de Santiago de Compostela

Autor: César Piñeiro Pomar

Tutores:

Juan Carlos Pichel Campos

José Manuel Abuín Mosquera

Julio 2016

Índice

- Motivación
- Introducción
- Objetivos
- Gestión del Proyecto
- Diseño
- Demostración
- Conclusiones
- Conocimiento Adquirido

Motivación

- Desarrollar un proyecto con una alta complejidad
- Complementar los conocimiento adquiridos en la asignatura de compiladores
- Traducir scripts Perl a Java de forma directa
- Generar código compatible con tecnologías Big Data
- Mejorar la experiencia de usuario

Introducción

- Perldoop
  - Compilador fuente-a-fuente Perl-Java
  - Programado en Python
- Etiquetas
  - Especificadas en comentarios
  - Añaden valor semántico
  - Aplican transformaciones
- Mejoras de la Versión 2.0
  - Herramientas para la construcción de compiladores
  - Simplificación de Etiquetas
  - Detección y gestión de errores

Objetivos

- Soporte para un subconjunto acotado de la sintaxis de Perl
- Conversión automática entre tipos de dato
- Traducir funciones personalizadas
- Funciones nativas implementadas
- Generar código paralelo compatible con Hadoop
- Gestión de errores

Gestión del proyecto

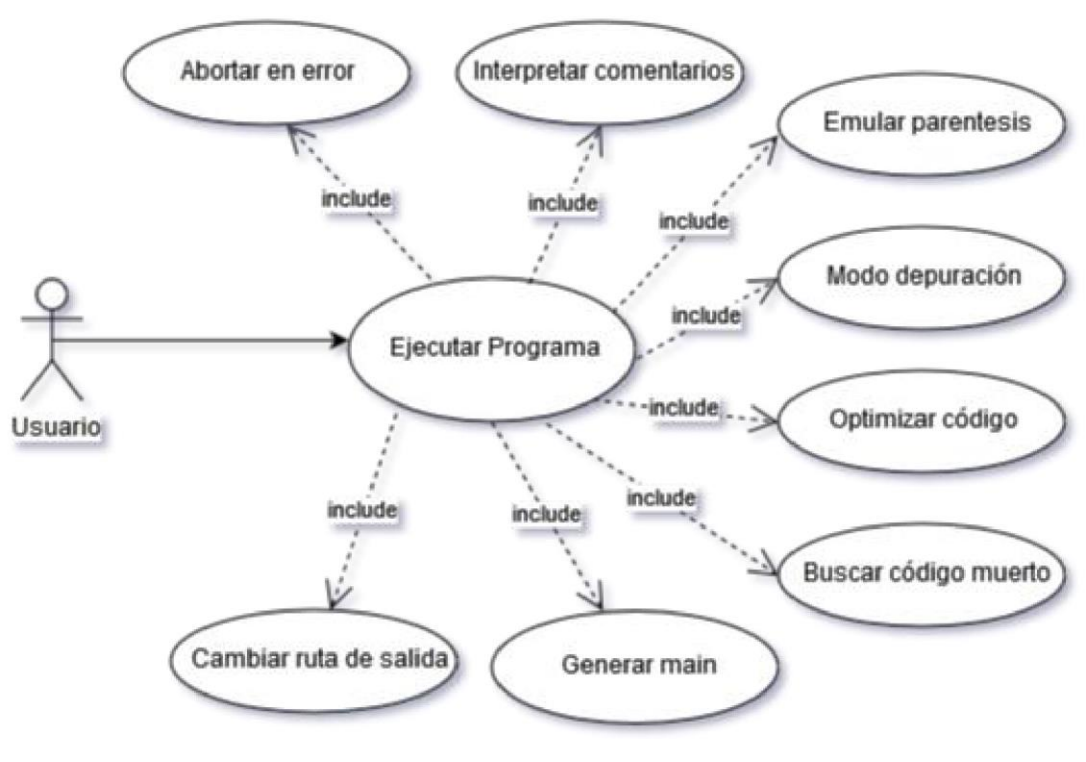
Gestión del Alcance

- Criterios de aceptación
  - Cumplir objetivos y requisitos
  - Aceptación por parte de los tutores
- Restricciones
  - El código debe contener etiquetas.
- Exclusiones
  - No es un traductor completo de Perl
  - No se incluye soporte para objetos
- Supuestos
  - Sintácticamente correcto
  - Semánticamente correcto

Gestión del proyecto

Casos de Uso

- Clasificados por importancia: Alta, Media, Baja.



Gestión del proyecto

Análisis de Requisitos

- Identificados
  - 18 Requisitos Funcionales: Interfaz por consola, Casting automatico…
  - 4 Requisitos no Funcionales: Portabilidad, Extensibilidad…
- Clasificados
  - Vitales: Generar código Hadoop
  - Importantes: Definir carpeta salida
  - Deseables: Traducir comentarios

Gestión del proyecto

Gestión de Riesgos

- Identificados
  - 3 Riegos en gestión de proyecto: Retraso en la planificación
  - 7 Riegos técnicos: Traducción incorrecta
- Escala impacto
  - Alto -> Impacto grave en la calidad y retraso importante planificación
  - Medio -> Impacto leve en la calidad y retraso leve planificación
  - Bajo -> Afecta a la planificación pero no retrasa la fecha de entrega
- Escala probabilidad
  - Alta -> Mayor o igual 70%
  - Media -> Entre 30% y 70%
  - Baja -> Menor o igual 30%

Gestión del proyecto

Metodología

- Características del proyecto
  - Proyecto individual
  - Componente de investigación
  - Requisitos Fijos
  - Fecha de entrega inamovible
- Programación Extrema
  - Metodología Ágil
  - Ciclo de desarrollo continuo
  - Código fuente como documentación

Gestión del proyecto

Planificación temporal

- Estructura descomposición de tareas (EDT)
  - Planificación y alcance: 1 semana
  - Formación: 1 semana
  - Análisis: 1 semana
  - Diseño: 1 semana y media
  - Implementación: 7 semanas
  - Pruebas: 1 semana y media
  - Documentación: 1 semana y media
- Cronograma
- Diagrama Gantt

Gestión del proyecto

Análisis de coste

- Estimación
  - Estación de trabajo
    - Ordenador Portátil: 1.800€
  - Software utilizado
    - Software libre: 0€
  - Repositorio Web
    - Bitbucket: 0€
  - Mano de Obra
    - 17,2 €/h sueldo medio ingeniero informático: 6.901,50€
- Total: 8.701,5€

Diseño

Etiquetas

- Importancia
  - Eliminar barreras entre Perl y Java
  - Evitar acciones por defecto
  - Aplicar transformaciones
- Definición
  - Dentro de comentarios
  - Encerrados entre < y >, ejemplo: <string>
- Usos
  - Añadir tipo a variables y funciones
  - Definir tamaño en las colecciones
  - Acotar código para ser paralelizado
  - Comportamiento léxicos

Diseño

Analizador Léxico y Sintáctico

- Léxico
  - Definición de tokens
  - Validar y ordenar de etiquetas
- Sintáctico
  - Ascendente
  - Gramática LALR(1)

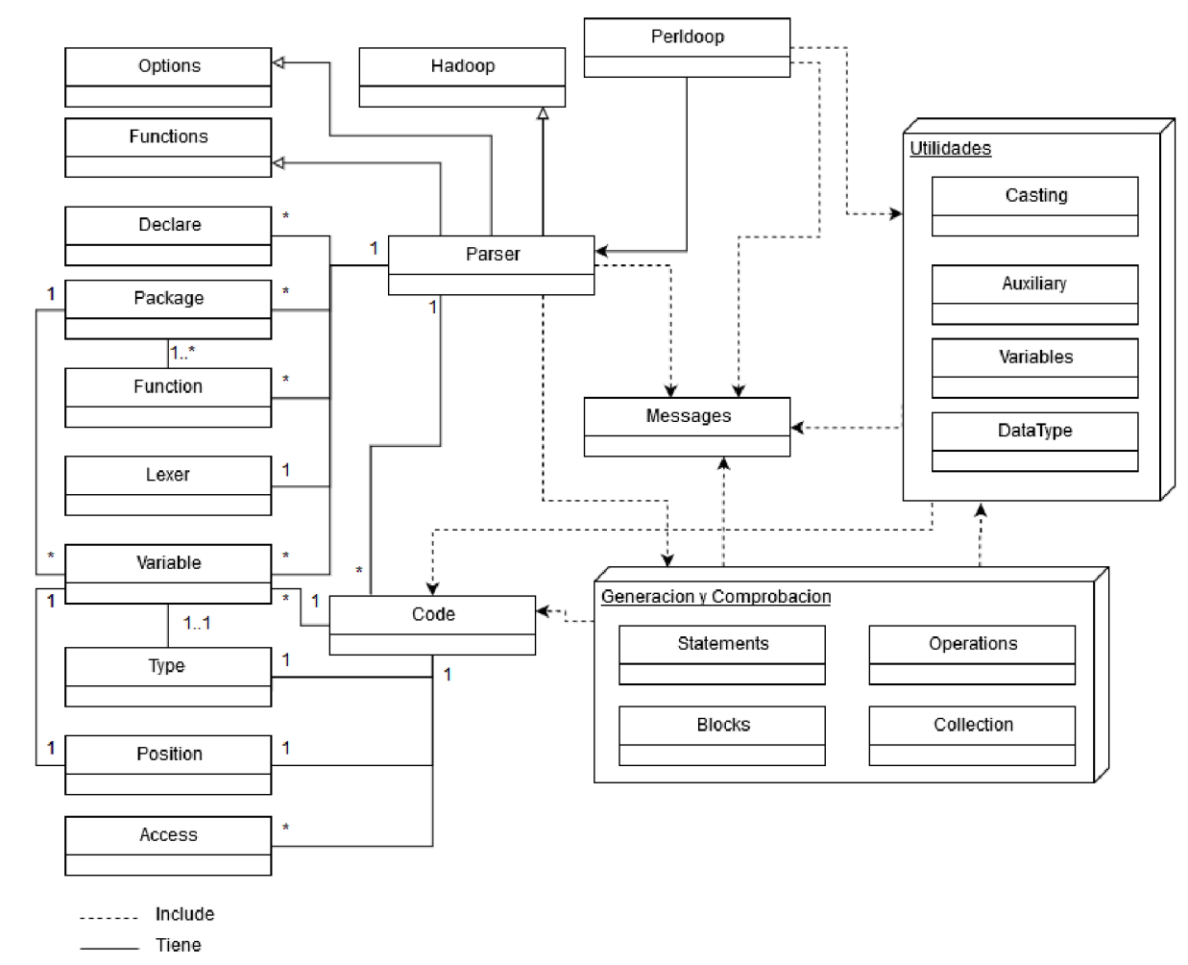
Diseño

Módulos

- Comprobación y generación de código
  - Una función por reducción sintáctica
  - Lanzan los errores
- Utilidades
  - Tipos de dato
  - Funciones de casting
  - Clases de almacenamiento
  - Funciones auxiliares
- Sistema de gestión de errores
  - Almacena los mensajes de error
  - Imprime el error por pantalla

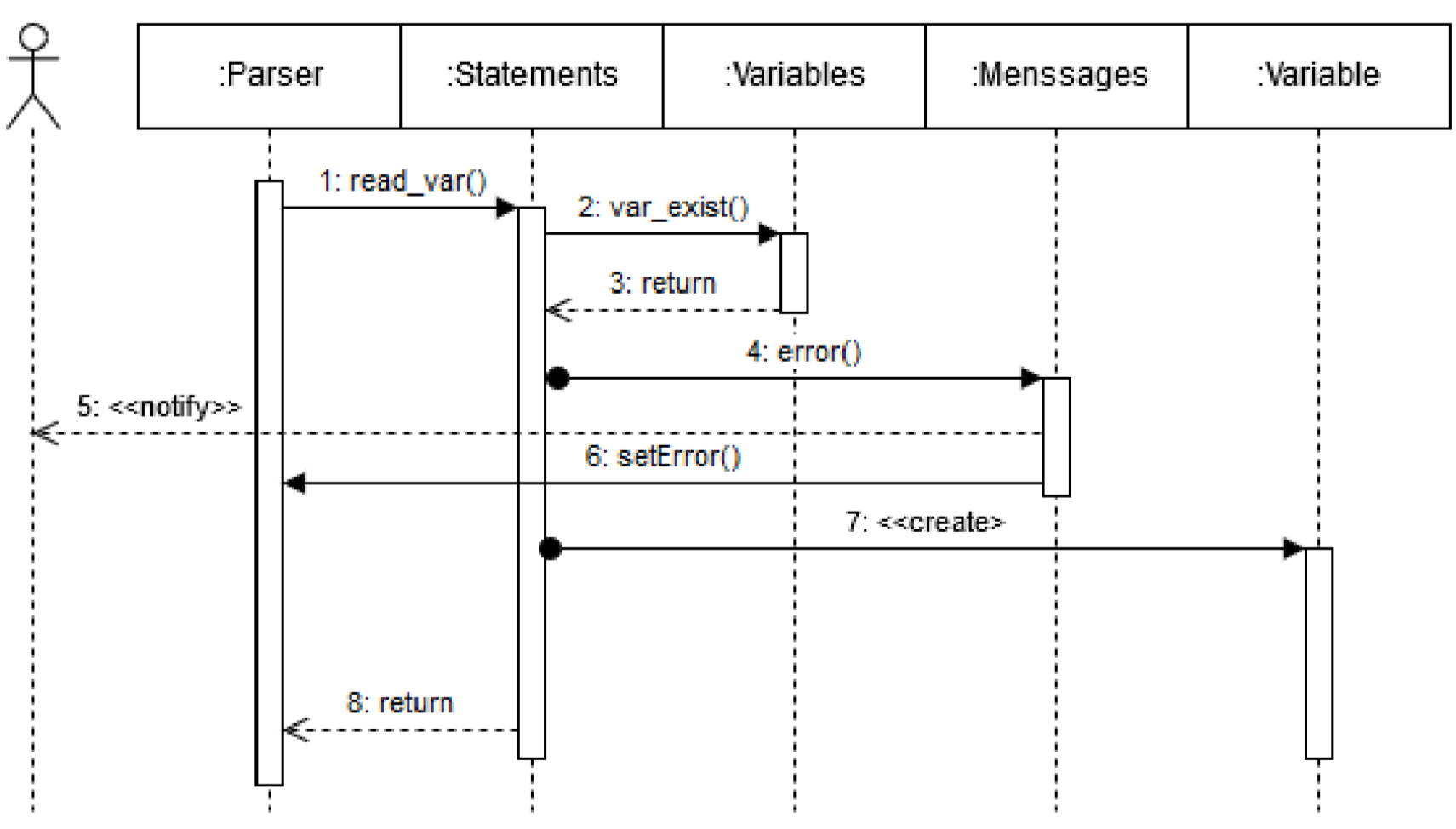
Diseño

Diagrama de clases



Diseño

Iteración durante un error



Diseño

Librería java

- Nivel de abstracción
- Simplificar traducción
- Tareas en tiempo de ejecución
- Funciones nativas

Demostración

- Interfaz consola
- Código Sumatorio
- Salida de error

```
cmd
<1> cmd
Microsoft Windows [Versión 10.0.10586]

César@CESAR-NITROV C:\Users\César
> perldoop -h
usage: perldoop.py [-h] [-m] [-out dir] [-c] [-ep] [-oc] [-uc] [-ea] [-dl]
                   [-dp] [-df file] [-ds size] [-dd]
                   infile [infile ...]

Perldoop 2.0: Un compilador fuente-a-fuente Perl-Java.

positional arguments:
  infile                Ficheros Perl para ser analizados.

optional arguments:
  -h, --help            show this help message and exit
  -m, --main            Crea una función de inicio "main" en el último fichero
                        para poder ejecutar el código directamente.
  -out dir              Carpeta donde se guardará los ficheros generados, por
                        defecto es el directorio actual.
  -c, --comments        Los comentarios dentro del código Perl, se mantendrán
                        en el código java.
  -ep, --emulate-parens
                        Añade automáticamente los paréntesis a las funciones,
                        si el código es sintácticamente correcto, debería
                        hacerlo correctamente.
  -oc, --optimize-code  Mejora el código de salida haciéndolo más visible y
                        eliminado redundancias dando lugar a un mayor
                        rendimiento.
  -uc, --unreachable-code
                        Comprueba la existencia de código muerto, si existe,
                        el código resultante no podrá ser copilado.
  -ea, --error-abort    Para el análisis en caso de encontrar un error.

cmd.exe*[64]:4820  « 160710[64]  1/1  [+] NUM  PRI↕  142x33  (3,51) 25V  8260  100%
```

Ejemplos

Archivo | Inicio | Compartir | Vista

Anclar al Acceso rápido | Copiar | Pegar | Cortar | Copiar ruta de acceso | Pegar acceso directo — Portapapeles

Mover a | Copiar a | Eliminar | Cambiar nombre — Organizar

Nueva carpeta | Nuevo elemento | Fácil acceso — Nuevo

Propiedades | Abrir | Modificar | Historial — Abrir

Seleccionar todo | No seleccionar ninguno | Invertir selección — Seleccionar

Ejemplos

Buscar en Ej...

- Acceso rápido
- OneDrive
- Equipo
- Red
- Grupo en el hogar

Perldoop.jar

Sumatorio.pl

2 elementos

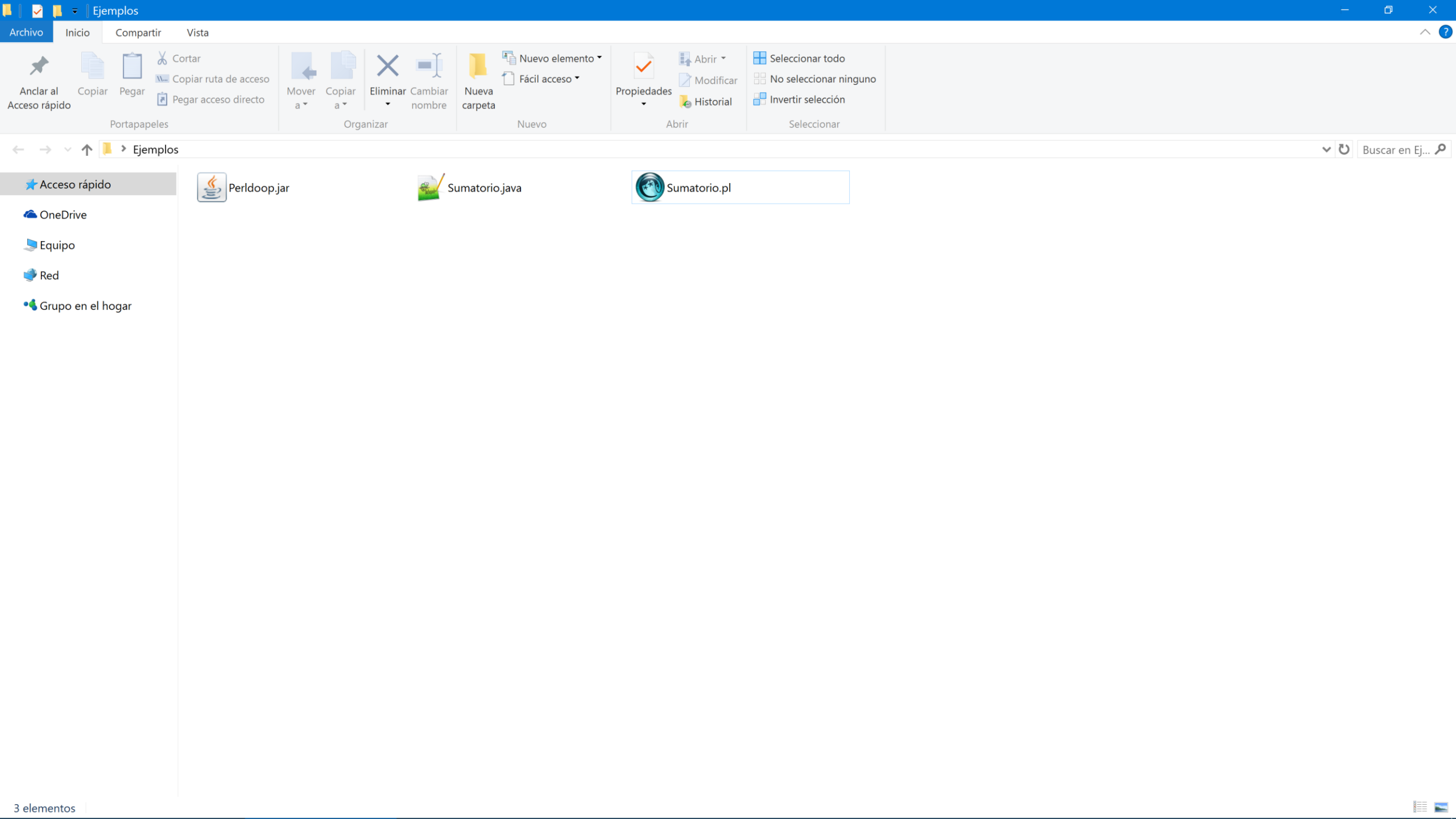
Sumatorio.pl

```
#<args><string>
#<returns><float>
sub sumatorio{
     (my $cadena,)=@_;#<string>
     my @numeros = split(",",$cadena);#<array><string>
     my $suma = 0;#<float>
     for my $n (@numeros){
          $suma += $n;
     }
     return $suma;
}

my $input = <STDIN>;#<string>
print sumatorio($input);
```

```
César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
> perl Sumatorio.pl
1,2,3,4,5,6,7,8,9
45
César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
> perldoop -m Sumatorio.pl

César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
>
```


Ejemplos
Archivo
Inicio
Compartir
Vista
Anclar al Acceso rápido
Copiar
Pegar
Cortar
Copiar ruta de acceso
Pegar acceso directo
Portapapeles
Mover a
Copiar a
Eliminar
Cambiar nombre
Organizar
Nueva carpeta
Nuevo elemento
Fácil acceso
Nuevo
Propiedades
Abrir
Modificar
Historial
Abrir
Seleccionar todo
No seleccionar ninguno
Invertir selección
Seleccionar
Ejemplos
Buscar en Ej...
Acceso rápido
OneDrive
Equipo
Red
Grupo en el hogar
Perldoop.jar
Sumatorio.java
Sumatorio.pl
3 elementos

C:\Users\César\Desktop\Ejemplos\Sumatorio.java - Notepad++

Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?

Sumatorio.pl | Sumatorio.java

```java
import perldoop.*;



public class Sumatorio{
    private static String[] ARGV;
    private static String input;


    public static Float sumatorio(Object... pd_argv){
        String cadena;
        cadena = ((String)pd_argv[0]);
        String[] numeros = Perl.split(",", cadena);
        Float suma = 0f;
        for(String n : numeros){
            suma = Pd.toFloat(suma + Double.parseDouble(n));
        }
        return suma;
    }

    public static void main(String[] ARGV){
        Sumatorio.ARGV=ARGV;
        input = Pd.read();
        Perl.print(sumatorio(input));
    }

}
```

Java source file | length : 628 | lines : 27 | Ln : 1 | Col : 1 | Sel : 0 | 0 | Dos\Windows | UTF-8 | INS

```
cmd
<1> cmd
Search

César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
> perl Sumatorio.pl
1,2,3,4,5,6,7,8,9
45
César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
> perldoop -m Sumatorio.pl

César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
> javac -cp ".;Perldoop.jar" Sumatorio.java

César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
> java -cp ".;Perldoop.jar" Sumatorio
1,2,3,4,5,6,7,8,9
45.0
César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
>

cmd.exe*[64]:4820  « 160710[64]  1/1  [+] NUM  PRI↕  142x33  (3,17) 25V  8260  100%
```

```
#<args><string>
#<returns><float>
sub sumatorio{
     (my $cadena,)=@_;#<string>
     my @numeros = split(",",$cadena);#<array><string>
     my $suma = 0;
     for my $n (@numeros){
          $suma += $n;
     }
     return $suma;
}

my $input = <STDIN>;#<string>
print sumatorio($input);
```

```
cmd
<1> cmd
Search

César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
> perldoop -m Sumatorio.pl
Sumatorio.pl:6:9: Error: La variable "suma" no tiene asignado un tipo
            my $suma = 0;
                ^

César@CESAR-NITROV C:\Users\César\Desktop\Ejemplos
>

cmd.exe*[64]:4820  « 160710[64]  1/1  [+] NUM  PRI↕  142x33  (3,9) 25V  8260  100%
```

Conclusiones

- La sintaxis de Perl es inmensa
- Su ambigüedad dificulta una traducción
- El tiempo es un limitante muy importante
- Características
  - Ficheros
  - Expresiones regulares
  - Módulos
  - Etc.
- Nuevas características pueden ser añadidas
  - Ampliar la sintaxis
  - Eliminar mas limitaciones
  - Mas funciones nativas
  - Nuevas tecnologías Big Data como Apache Spark

Conocimiento Adquirido

- Conocimiento y soltura con Python y Perl
- Manejo mas profundo de la creación de gramáticas
- Diseño de un compilador con todas sus fases
- Experiencia en tecnologías Big Data con Hadoop