

Procesos ETL

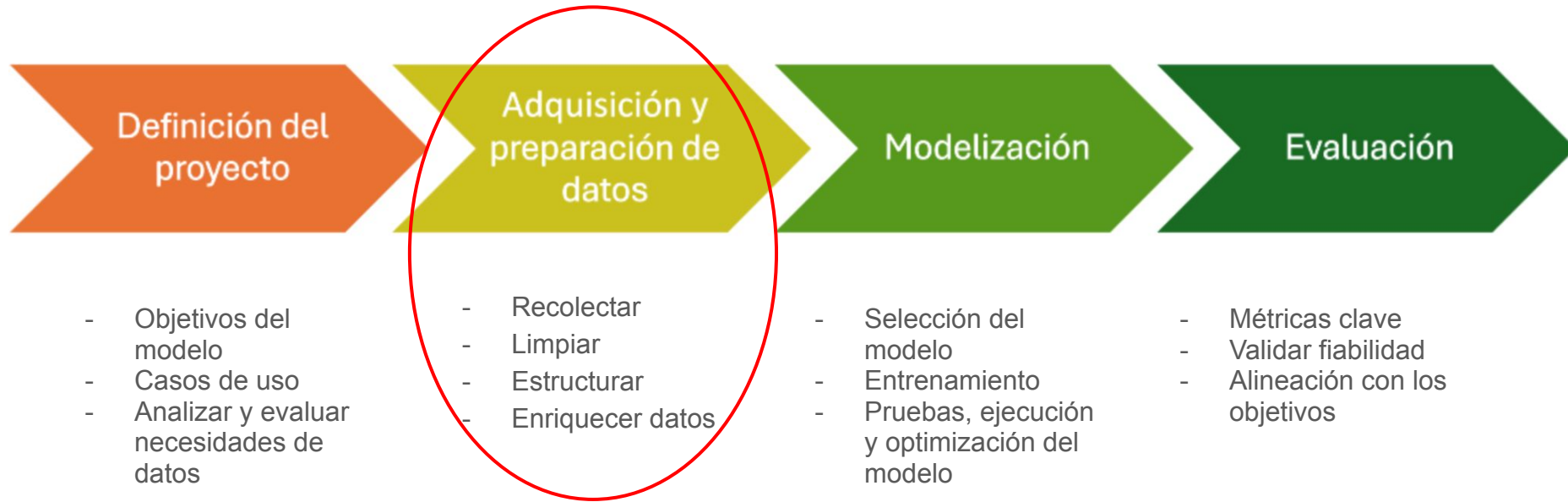
Tema 6: Limpieza, transformación y normalización de datos. Parte I Expresiones regulares

Adquisición y preparación de datos

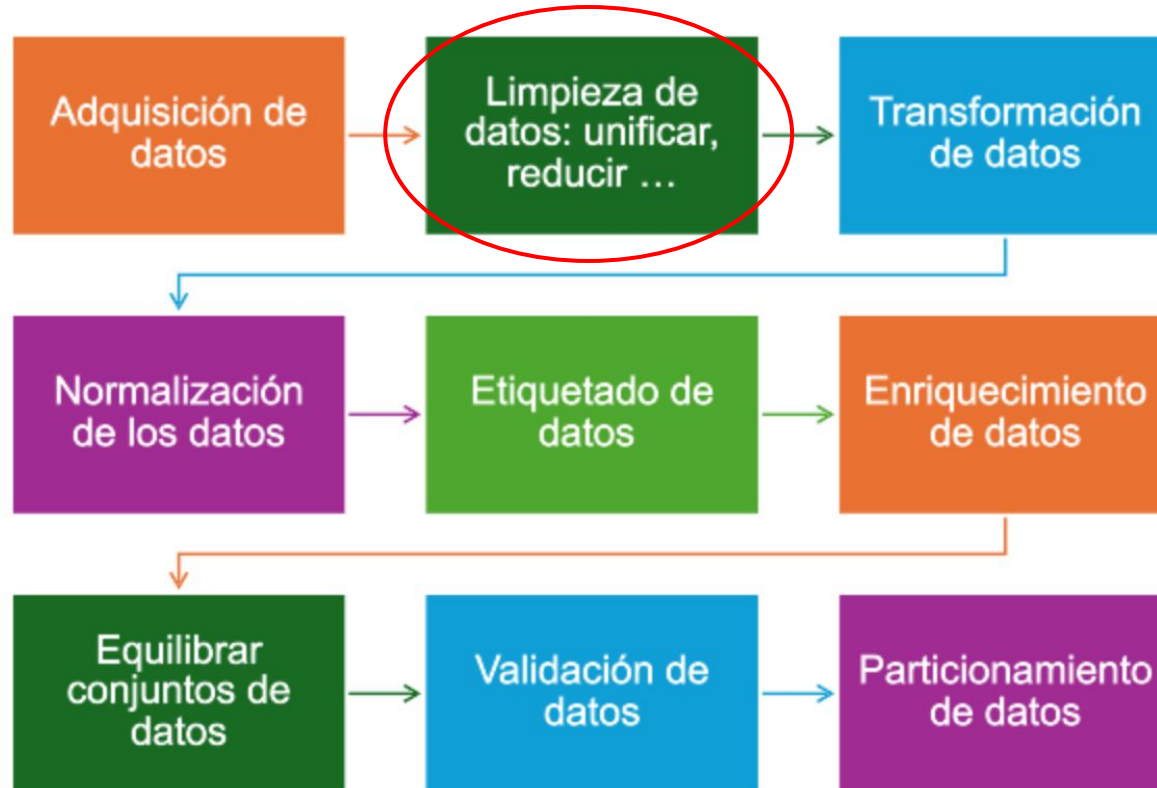


Universitat d'Alacant
Universidad de Alicante

Ciclo de vida del proyecto IA



Preparación de datos



Qué es un proceso ETL

Extract

- **Extraer** datos desde las distintas fuentes de origen - fusión de datos
 - Identificar fuentes
 - Identificar conjunto de datos
 - Conectores para la adquisición
 - Fuentes como Hadoop, datos abiertos, Sql, Nosql (MongoDB, Cassandra), repositorios RDF, Wikidata, etc
- **Analizar** los datos obtenidos
 - Tipos de datos
 - Codificación origen
 - Semántica
- **Interpretar** y comprobar que cumplen con la estructura esperada

Qué es un proceso ETL

Transform

Depurar y ajustar a los requisitos, aplicando **reglas** que guíen las transformaciones de los datos para un formato final, contenido y forma

- Selección de datos (columnas, filas...), eliminado ruido
- Modificación del contenido o la estructura de los datos
- Integración con datos de otras fuentes
- Corrección de errores
- Cálculo de valores derivados o agregados a partir de datos procesados
- Traducción de valores
- Codificación
- Tratamiento de textos
- Validación de datos frente a reglas de calidad

Qué es un proceso ETL

Load

Carga y actualización de los datos en el almacén de datos

- Selección sistema destino: fichero, *data warehouse*, BD relacional, índice, NoSql, etc
- Nuevo o sobrescribir (ejemplo BD, Excel)
- Historial de cambios

Tipos de proceso de carga:

- **Acumulación simple**
- **Rolling** para mayor granularidad

Limpieza de datos

Limpieza de datos

Los datos sin procesar suelen ser imperfectos, contienen errores, faltan valores o están duplicados.

Objetivo de la limpieza de datos es **eliminar** estas **anomalías** para **garantizar la calidad y su fiabilidad**.

Una limpieza adecuada ayuda a evitar que los datos defectuosos afecten al rendimiento del modelo.

- Corregir **errores**
- Eliminar **duplicados**
- Gestionar valores que **faltan**: reemplazo, interpolación o eliminación
- Gestionar valores **atípicos**
- **Unificar** de tipos de datos
- Aplicar la **reducción** de datos
 - La reducción de datos filtra los campos que no son necesarios
 - Eliminar los valores que se encuentran fuera del rango esperado
 - Redundancia, conservar datos redundantes puede ralentizar el análisis y consumir recursos

Limpieza de datos

Limpieza de datos mediante:

1. Expresiones regulares

- Introducción
- Sintaxis
- Ejemplos

2. Otros mecanismos para la limpieza de datos

Expresiones regulares

Expresiones regulares

Mecanismo **eficiente y flexible** que facilita el procesamiento de datos.

Secuencia de caracteres que forman un patrón que facilitan **filtrar, encontrar, validar y tratar** un texto.

Sus principales componentes:

- **Literales**: caracteres que conforman cualquier palabra
- **Clases** de caracteres: son definidas a través de una lista de caracteres entre corchetes
- **Metacaracteres**: elemento especial que permiten delimitar, iterar, alternar a los literales.

Regular Expressions Cheat Sheet by
Dave Child (DaveChild) via
cheatography.com/1/cs/5/

Cheatography Regular Expressions Cheat Sheet
by Dave Child (DaveChild) via cheatography.com/1/cs/5/

By **Dave Child** (DaveChild) cheatography.com/davechild/ www.addedbytes.com Published 19th October, 2011. Last updated 2nd January, 2015. Page 1 of 1. Sponsored by **Readability-Score.com** Measure your website readability! <https://readability-score.com>

Expresiones regulares

Consultar RegExLib.com Regular Expression Cheat Sheet

regexlib.com/CheatSheet.aspx

RegExLib.com Regular Expression Library		RegExLib.com Regular Expression Cheat Sheet (.NET)	
Metacharacters Defined		Metacharacter Examples	
^	Start of a string.	^abc	abc, abcdefg, abc123, ...
\$	End of a string.	abc\$	abc, endsinabc, 123abc, ...
.	Any character (except \n newline)	a.c	abc, aac, acc, adc, aec, ...
	Alternation.	bill ted	ted, bill
{...}	Explicit quantifier notation.	ab(2)c	abbc
[...]	Explicit set of characters to match.	a[bb]c	abc, a8c
(...)	Logical grouping of part of an expression.	(abc){2}	abcbac
*	0 or more of previous expression.	ab*c	ac, abc, abbc, abbbc, ...
+	1 or more of previous expression.	ab+c	abc, abbc, abbbc, ...
?	0 or 1 of previous expression; also forces minimal matching when an expression might match several strings within a search string.	ab?c	ac, abc
\	Preceding one of the above, it makes it a literal instead of a special character. Preceding a special matching character, see below.	a\sc	a c

Character Escapes http://tinyurl.com/5wm3wl	
ordinary characters	Characters other than . \$ ^ { [] } * + ? \ match themselves.
\a	Matches a bell (alarm) \u0007.
\b	Matches a backspace \u0008 if in a []; otherwise matches a word boundary (between \w and \W characters).
\t	Matches a tab \u0009.
\r	Matches a carriage return \u000D.
\v	Matches a vertical tab \u000B.
\f	Matches a form feed \u000C.
\n	Matches a new line \u000A.
\e	Matches an escape \u001B.
\040	Matches an ASCII character as octal (up to three digits); numbers with no leading zero are backreferences if they have only one digit or if they correspond to a capturing group number. (For more information, see Backreferences.) For example, the character \040 represents a space.
\x20	Matches an ASCII character using hexadecimal representation (exactly two digits).
\cC	Matches an ASCII control character; for example \cC is control-C.
\u0020	Matches a Unicode character using a hexadecimal representation (exactly four digits).
*	When followed by a character that is not recognized as an escaped character, matches that character. For example, * is the same as \x2A.

Character Classes http://tinyurl.com/5ck4ll	
.	Matches any character except \n. If modified by the Singleline option, a period character matches any character. For more information, see Regular Expression Options.
[aeiou]	Matches any single character included in the specified set of characters.
[^aeiou]	Matches any single character not in the specified set of characters.
[0-9a-f-F]	Use of a hyphen (-) allows specification of contiguous character ranges.
\p{name}	Matches any character in the named character class specified by {name}. Supported names are Unicode groups and block ranges. For example, Ll, Nd, Z, IsGreek, IsBoxDrawing.
\P{name}	Matches text not included in groups and block ranges specified in {name}.
\w	Matches any word character. Equivalent to the Unicode character categories [\p{L}]\p{Lu}]\p{Ll}]\p{Lo}]\p{Nd}]\p{Pc}]. If ECMAScript-compliant behavior is specified with the ECMAScript option, \w is equivalent to [a-zA-Z_0-9].
\W	Matches any nonword character. Equivalent to the Unicode character categories [^\p{L}]\p{Lu}]\p{Ll}]\p{Lo}]\p{Nd}]\p{Pc}]. If ECMAScript-compliant behavior is specified with the ECMAScript option, \W is equivalent to [^a-zA-Z_0-9].
\s	Matches any white-space character. Equivalent to the Unicode character categories [\p{n}]\p{t}]\p{x85}]\p{Z}]. If ECMAScript-compliant behavior is specified with the ECMAScript option, \s is equivalent to [\p{n}]\p{t}]\p{v}].
\S	Matches any non-white-space character. Equivalent to the Unicode character categories [^\p{n}]\p{t}]\p{x85}]\p{Z}]. If ECMAScript-compliant behavior is specified with the ECMAScript option, \S is equivalent to [^\p{n}]\p{t}]\p{v}].
\d	Matches any decimal digit. Equivalent to \p{Nd} for Unicode and [0-9] for non-Unicode, ECMAScript behavior.

Expresiones regulares

Algunos de los más utilizados *metacharacters*:

- \ Indica que el siguiente carácter es especial y se desea tratar como un literal
- ^ Indica el principio de una cadena
- \$ Indica el final de una cadena
- . Cualquier carácter salvo el salto de línea (\n)
- (...) Agrupación lógica de partes de una expresión.
- [...] Un conjunto de caracteres de la expresión
 - [xyz] Indica coincidencia con cualquiera de los caracteres entre corchetes
 - [^xyz] cualquiera carácter que no esté entre corchetes
 - [a-z] cualquier carácter dentro del rango especificado
 - [0-9a-fA-F] cualquier carácter dentro de los rangos especificados
- {...} Indica un número o intervalo de longitud de la expresión
 - {n} repetición de un carácter n veces.
 - {n,} repetición de un carácter como mínimo n veces
 - {n,m} repetición de un carácter como mínimo n veces y máximo m veces
- ? 0-1 ocurrencias de la expresión
- + 1-n ocurrencias de la expresión
- * 0-n ocurrencias de la expresión
- | Para indicar una disyunción lógica, elegir entre dos valores x|y se tiene que cumplir al menos uno de los dos

Expresiones regulares

Algunos de los más utilizados *character classes*:

- **\w** Coincide con cualquier carácter alfanumérico o `_`, equivale a `[a-zA-Z_0-9]`
- **\W** Coincide con cualquier carácter no alfanumérico ni `_`, equivale a `[^a-zA-Z_0-9]`
- **\s** Espacio en blanco, ECMAScript equivale a `[\f\n\r\t\v]`
- **\S** No espacio en blanco, ECMAScript equivale a `[^\f\n\r\t\v]`
- **\d** Dígito decimal, equivale a `[0-9]`
- **\D** No dígito decimal, equivale a `[^0-9]`
- **\x** Dígito hexadecimal
- **\p{name}** Coincide con cualquier carácter de la clase o categoría Unicode especificada por {name}.
- **\P{name}** Coincide con el texto no incluido en los grupos y rangos de bloques especificados en {name}

Anchors:

- **\b** Límite de palabra
- **\B** No límite de palabra

Expresiones regulares

Flags, modifican el comportamiento del patrón

- **g** encuentra todas las coincidencias, si no se usa solo devuelve la primera coincidencia
- **i** no distingue entre mayúsculas y minúsculas
- **m** Para múltiples líneas
- **s** Habilita el modo “dotall”, un punto “.” coincide con el carácter de línea nueva \n tratando a la cadena como una única línea
- **y** búsqueda en la posición exacta del texto
- **u** Permite el soporte completo de Unicode

Expresiones regulares


PDI incluye pasos para el uso de expresiones regulares.

Facilita la inclusión de scripts en Javascript para la limpieza de datos y otras tareas.

En JavaScript se pueden utilizar los métodos `exec()` y `test()` de `RegExp`, y `match()`, `matchAll()`, `replace()`, `replaceAll()`, `search()` y `split()` de `String`.

```
str = "Impreso en casa de Monreal núm. 2, 1902";  
const regex = /\d+$/;  
console.log(regex.test(str)); // true
```

```
console.log(regex.exec(str));
```



```
[  
  '1902',  
  index: 35,  
  input: 'Impreso en casa de Monreal núm. 2, 1902',  
  groups: undefined  
]
```

```
const regex2 = /\d+/g;  
const result = Array.from(str.matchAll(regex2), (m) => m[0]);
```

```
console.log(result);
```



```
[ '2', '1902' ]
```

Expresiones regulares

^ \$ Inicio o final de un string (*anchors*)

- `^(.+:)` o bien `^(Auxiliar localización):?()?`

```
s = "Auxiliar localización: Parque Natural de Redes, Asturias";  
const pRE = /^(Auxiliar localización):?( )?.*;/;  
pRE.test(s);
```

- `(.*)((\d+))$` cadena que termine en número

```
s = "Auxiliar localización: Parque Natural de Redes, Asturias, 1968";  
const pRE = /.*((\d+))$/;  
pRE.test(s);
```

- Ambos se puede usar juntos para comprobar si una cadena coincide completamente con el patrón

```
s = "09:37";  
const pRE = /^(\d):(\d):(\d)$/;  
pRE.test(s);
```

Expresiones regulares

- **\b** Búsqueda de palabras completas

```
s1 = "Edición digital a partir de Obras completas de Baltasar Gracián. Vol II";
s2 = "Ed. Emilio Blanco sin Volumen definido. 1993";

const pRE = /\bVol\b/;
pRE.test(s1);
pRE.test(s2);

console.log(s1.replace(pRE, 'Volumen'));
console.log(s2.replace(pRE, 'Volumen'));

const pRE2 = /Vol/;
pRE2.test(s1);
pRE2.test(s2);

console.log(s1.replace(pRE2, 'Volumen'));
console.log(s2.replace(pRE2, 'Volumen'));
```

Expresiones regulares

- \ Indica que el siguiente carácter es especial y se desea tratar como un literal
 - Para escapar caracteres especiales [\ ^ \$. | ? * + ()
 - Se debe escapar / si se usa el patrón /.../ (pero no dentro de new RegExp).
 - Si se usa new RegExp, se deben duplicar las barras invertidas \, porque las comillas de cadena consumen una.

```
s1 = "Edición digital a partir de Obras completas de Baltasar Gracián. Volumen II";  
s2 = "Ed. Emilio Blanco sin Volumen, (Murcia)";  
  
const pRE = /\./;  
pRE.test(s1);  
  
const pRE2 = /(.*,)?\((?\.+)\)?$/;  
pRE2.test(s1);  
pRE2.test(s2);
```

Expresiones regulares

- **()** Para agrupar

```
s1 = "Auxiliar localización: Sanlúcar de Barrameda, Cádiz";  
s2 = "Ed. Emilio Blanco sin Volumen, (Murcia)";  
  
const pRE = /(.*,)?\((?.+)\)?$/;  
pRE.test(s1); // → true grupo2 = Cádiz  
pRE.test(s2); // → true grupo2 = Murcia
```

Expresiones regulares

- [...] Un conjunto de caracteres de la expresión
 - [xyz] Indica coincidencia con cualquiera de los caracteres entre corchetes
 - [^xyz] cualquiera caracter que no esté entre corchetes
 - [a-z] cualquier caracter dentro del rango especificado
 - [0-9a-fA-F] cualquier caracter dentro de los rangos especificados

```
//Obtener todas las obras cuya reproducción indique un volumen. Existen diferentes formas de indicarlo
```

```
s1 = "Edición digital a partir de Obras completas de Baltasar Gracián. Volumen II";
```

```
s2 = "Ed. Emilio Blanco volumen I, pp23, (Murcia)";
```

```
s3 = "Ed. Emilio Blanco Vol.2 (Murcia)";
```

```
const pRE = /[Vv]olumen/;
```

```
pRE.test(s1);
```

```
pRE.test(s2);
```

```
pRE.test(s3);
```

```
const pRE2 = /[Vv]ol(\.|umen)/;
```

```
pRE2.test(s1);
```

```
pRE2.test(s2);
```

```
pRE2.test(s3);
```

Expresiones regulares

```
//Obtener todas las obras cuya reproducción indique un volumen. Existen diferentes formas de indicarlo

s1 = "Edición digital a partir de Obras completas de Baltasar Gracián. Volumen II";
s2 = "Ed. Emilio Blanco volumen I, pp23, (Murcia)";
s3 = "Ed. Emilio Blanco Vol.2 (Murcia)";

const pRE = /[Vv]olumen/;
pRE.test(s1);
pRE.test(s2);
pRE.test(s3);

const pRE2 = /[Vv]ol(\.|umen)/;
console.log(pRE2.test(s1));
console.log(pRE2.test(s2));
console.log(pRE2.test(s3));
```

¿Alguna otra expresión regular que podamos usar?

Expresiones regulares

- [...] Un conjunto de caracteres de la expresión

Las clases de caracteres son abreviaturas para ciertos conjuntos de caracteres:

`\d` \rightarrow `[0-9]`

`\w` \rightarrow `[a-zA-Z0-9_]`

`\s` \rightarrow `[\t\n\v\f\r]`, además de otros caracteres de espacio de unicode.

Expresiones regulares

Ejercicio 1 - Obtener expresión regular para comprobar que una fecha del siglo 21 es correcta en formato YYYY-MM-DD

```
s1 = "[Vicente Barberá Masip o Enrique Desfilis Barberá]., 1898";  
s2 = "Imp. Vda. de Ayoldi, 1883-06-24";  
s3 = "Direcció General del Llibre, Arxius i Biblioteques, 2022-12-31";
```

```
const pRE = /.../;
```



```
console.log(pRE.test(s1));  
console.log(pRE.test(s2));  
console.log(pRE.test(s3));
```

Expresiones regulares

- . Cualquier carácter salvo el salto de línea
- + 1-n ocurrencias de la expresión
- * 0-n ocurrencias de la expresión
- ? 0-1 ocurrencias de la expresión

```
s1 = "Auxiliar localización: Sanlúcar de Barrameda, Cádiz";  
s2 = "Ed. Emilio Blanco sin Volumen, (Murcia)";  
s3="";
```

```
pRE = /.*/;  
console.log(pRE.test(s1)); // → true  
console.log(pRE.test(s3)); // → ?
```

```
pRE = /.+/  
console.log(pRE.test(s3)); // → ?
```

```
pRE = /(.,)?\?(?(\.+)?)?$/;  
pRE.test(s1); // → ?  
pRE.test(s2); // → ?
```

Expresiones regulares

- **{...}** Indica un número o intervalo de longitud de la expresión
 - {n} repetición de un carácter n veces.
 - {n,} repetición de un carácter como mínimo n veces
 - {n,m} repetición de un carácter como mínimo n veces y máximo m veces

```
codigos = "CP: 28001, 03001, 123456.";
console.log(codigos.match(/\b\d{5}\b/g));

//Extrae número de registro variable
idReg = "ID123, ID4567, ID7450123.";
validados = idReg.match(/\d{4,6}/g);
```

Expresiones regulares

Ejercicio 2 - Validar una contraseña mediante una expresión regular. La contraseña debe tener entre 8 y 16 caracteres, contener al menos una letra minúscula, una mayúscula y un número.

Expresiones regulares

Ejercicio 3 - ¿Cómo limpiamos dos o más espacios en blanco en la siguiente cadena de texto?

Es el título original de una obra publicada en la BVMC, tal cual se puede leer a continuación:

Recopilacion verificada por Felipe Mateu, de las noticias y reglas por las que se gobierna la Real Acequia de Moncada (Valencia), efectuada por acuerdo de su Junta de cequeiros en 10 de octubre de 1757 : Concluida esta recopilación en 16 de julio de 1758

Expresiones regulares

Ejercicio 3:

```
s1 = "Recopilacion verificada por Felipe Mateu, de las noticias y reglas por las que se gobierna la  
Real Acequia de Moncada (Valencia), efectuada por acuerdo de su Junta de cequeiros en 10 de octubre de  
1757 : Concluida esta recopilación en 16 de julio de 1758";
```

```
const pRE = /( ){2,}/;  
const pRE2 = /\s{2,}/;  
const pRE3 = /\s+/g;  
const pRE4 = /\s{2,}/g;
```

```
console.log(pRE.test(s1));  
console.log(pRE2.test(s1));  
console.log(pRE3.test(s1));  
console.log(pRE4.test(s1));
```

```
let s1SinEspaciosMultiples = s1.replace(/\s+/g, '');
```

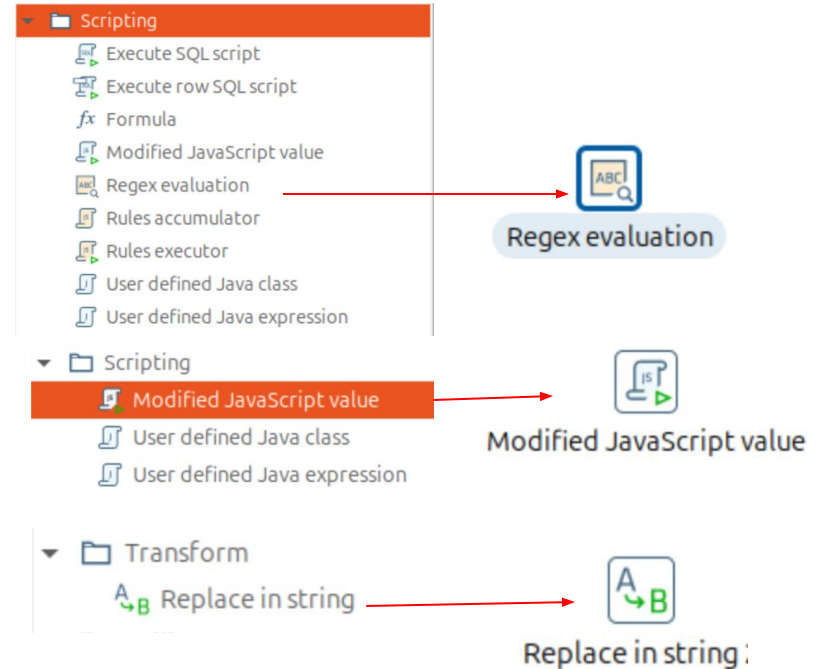
```
console.log(pRE4.test(s1SinEspaciosMultiples));
```



Expresiones regulares en PDI

Ejemplos de uso en Pentaho Data Integration

- Reemplazar partes en una cadena de texto ***replace string***
- Obtener datos de un texto mediante **regex evaluation**
- Expresiones regulares **REGEX** en MySQL
- **JavaScript**



Expresiones regulares en PDI

Replace string

Se debe limpiar los campos de unos registros bibliográficos obtenidos de una fuente externa.

Después de analizar una muestra, se determina que es necesario adecuar a la norma de catalogación y forma de descripción correcta para el almacén de datos final.

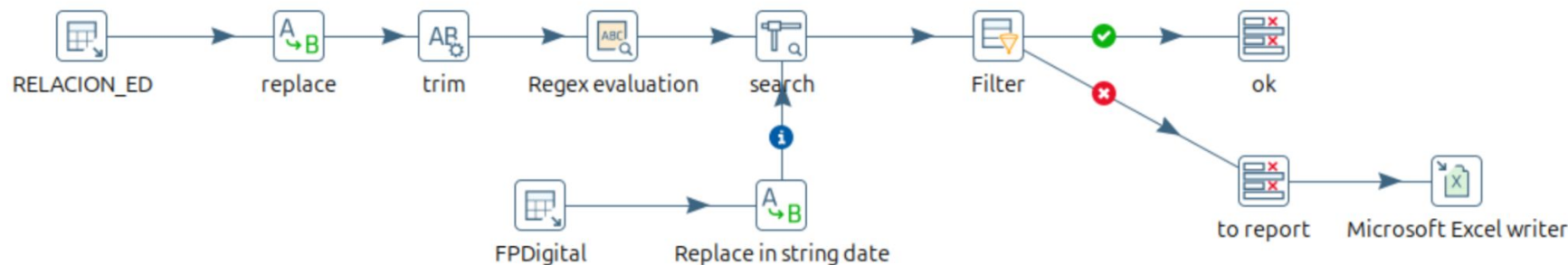
Para ello se solicita realizar un proceso ETL para dicha tarea.

encionPublicacion

"Cuadernos Hispanoamericanos", núm. 91-92 (julio-agosto 1957), pp. 180-201
"Cuadernos Hispanoamericanos", núm. 45 (septiembre 1953), pp. 279-291
"Cuadernos Hispanoamericanos", núm. 22 (julio-agosto 1951), pp. 15-19
"Cuadernos Hispanoamericanos", núm. 26 (febrero 1952)
"Cuadernos Hispanoamericanos", núm. 20 (marzo-abril 1951), pp. 169-185
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 237-238
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 241-242
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 243-286
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 289-299
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 301-306
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 323-332
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 335-381
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 383-397
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 399-414
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 415-417
"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 421-426

Expresiones regulares en PDI

Replace string



757355	"Cuadernos Hispanoamericanos", núm. 91-92 julio-agosto 1957	"Cuadernos Hispanoamericanos", núm. 91-92 (julio-agosto 1957), pp. 180-201
782746	"Cuadernos Hispanoamericanos", núm. 45 septiembre 1953	"Cuadernos Hispanoamericanos", núm. 45 (septiembre 1953), pp. 279-291
970729	"Cuadernos Hispanoamericanos", núm. 22 julio-agosto 1951	"Cuadernos Hispanoamericanos", núm. 22 (julio-agosto 1951), pp. 15-19
972802	"Cuadernos Hispanoamericanos", núm. 26 febrero 1952	"Cuadernos Hispanoamericanos", núm. 26 (febrero 1952)
975986	"Cuadernos Hispanoamericanos", núm. 20 marzo-abril 1951	"Cuadernos Hispanoamericanos", núm. 20 (marzo-abril 1951), pp. 169-185
984341	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 237-238
984389	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 241-242
984392	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 243-286
984397	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 289-299
984401	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 301-306
984407	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 323-332
984413	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 335-381
984449	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 383-397
984452	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 399-414
984455	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 415-417
984461	"Cuadernos Hispanoamericanos", núm. 11-12 septiembre-diciembre 1949	"Cuadernos Hispanoamericanos", núm. 11-12 (septiembre-diciembre 1949), pp. 421-426

Expresiones regulares en PDI

Replace string

Replace in string

Step name

replace

Fields string

▲	In stream field	O	use RegEx	Search	Replace with	Set empty string?	Replace with field	Whole Word	Case sensitive	Is Unicode
1	lugarPublicacionOriginal		Y	, pp\..*\$		N		N	N	N
2	lugarPublicacionOriginal		Y	, p\..*\$		N		N	N	N
3	lugarPublicacionOriginal		Y	(\ \\ ?)\$		N		N	N	N
4	lugarPublicacionOriginal		Y	(\ \\ ?)\$		N		N	N	N
5	lugarPublicacionOriginal		Y	(\\ \\)		N		N	N	N
6	lugarPublicacionOriginal		Y	(\\-[0-9]{1,2})\\-[0-9]{1,2})\$		N		N	N	N

Help

OK

Get fields

Cancel

Expresiones regulares en PDI

String operations

String operations

Step name

The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape ▲	Digits	Remove Special character
1	lugarPublicacionOriginal		both								carriage return & line feed
2											

[Help](#)[OK](#)[Get fields](#)[Cancel](#)

Expresiones regulares en PDI

Regex evaluation

Extraer fecha de publicación de la nota de mención original de una obra.

Debe estar en formato YYYY

Step name: **Regex evaluation**

Settings | Content

Step settings

Field to evaluate: lugarPublicacionOriginal

Result field name: result

Create fields for capture groups: ☒

Replace previous fields: ☒

Regular expression: `(.* |, |ca\\. |.*) ([0-9]{4})$` Test regex

Use variable substitution: ☐

Capture Group Fields

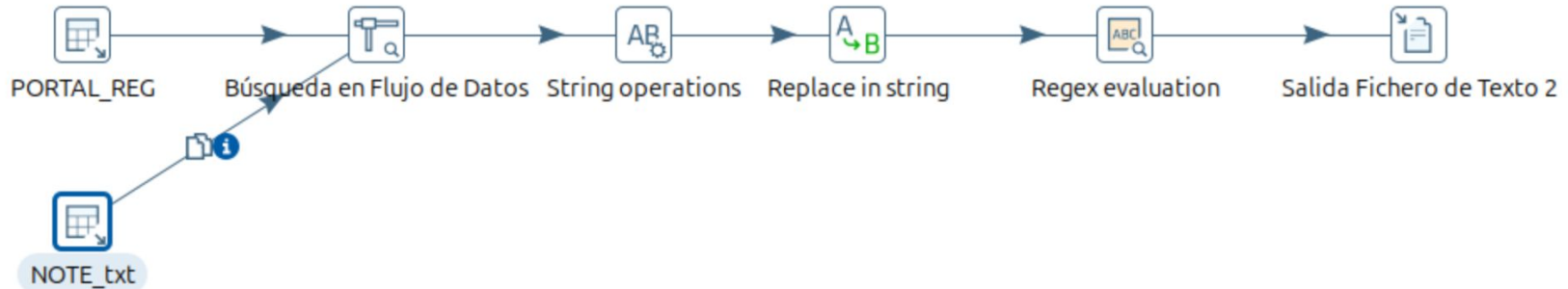
	New field	Type	Length	Precision	Format	Group	Decimal
1	aux	String					
2	dateLPO	String					

Help OK Cancel

Expresiones regulares en PDI

Regex evaluation

Regex evaluation para evaluar el contenido de una nota de texto. Obtiene diferentes partes significativas de contenido para esta tarea



Expresiones regulares en PDI

Regex evaluation

Step name: Regex evaluation

Settings Content

Step settings

Field to evaluate: nota

Result field name: result

Create fields for capture groups: ☒

Replace previous fields: ☒

Regular expression: ☐ Use variable substitution

Test regEx

Capture Group Fields

#	New field	Type	Length	Precision	Format	Group	Decimal	Currency	Null if	Default	Trim
1	local	String									both
2	provincia	String									none

Help OK Cancel

Regular expression evaluation

Regular expression

Please enter a new regular expression or modify.

(.*,)?(?{.+})??

The regular expression was successfully compiled.

Values to test

Value1

Value2

Value3

Capture

Capture from value: una localización: Santiago de la Rivera, Murcia

Capture group fields (2): una localización: Santiago de la Rivera, Murcia

OK Cancel

Expresiones regulares en PDI

`(. *,)?(?(.+)\)?$`

Examine preview data				
	notaOriginal	result	local	provincia
	Auxiliar localización: Madrid	Y	<null>	Madrid
	Auxiliar localización: Salnés, Pontevedra	Y	Salnés,	Pontevedra
	Auxiliar localización: Albaicín, Granada	Y	Albaicín,	Granada
	Auxiliar localización: Granada	Y	<null>	Granada
	Auxiliar localización: Albaicín, Granada	Y	Albaicín,	Granada
	Auxiliar localización: Albaicín, Granada	Y	Albaicín,	Granada
	Auxiliar localización: Albaicín, Granada	Y	Albaicín,	Granada
	Auxiliar localización: Granada	Y	<null>	Granada
	Auxiliar localización: Granada	Y	<null>	Granada
	Auxiliar localización: Granada	Y	<null>	Granada
	Auxiliar localización: Granada	Y	<null>	Granada
	Auxiliar localización: Granada	Y	<null>	Granada
	Auxiliar localización: Motril, Granada	Y	Motril,	Granada
	Auxiliar localización: Sanlúcar de Barrameda, Cádiz	Y	Sanlúcar de Barrameda,	Cádiz
	Auxiliar localización: A Coruña	Y	<null>	A Coruña
	Auxiliar localización: Toledo	Y	<null>	Toledo
	Auxiliar localización: Jaén	Y	<null>	Jaén
	Auxiliar localización: Córdoba	Y	<null>	Córdoba
	Auxiliar localización: Palencia	Y	<null>	Palencia
	Auxiliar localización: Tuñón, Asturias	Y	Tuñón,	Asturias
	Auxiliar localización: Olot, Girona	Y	Olot,	Girona

Expresiones regulares en PDI

Funciones y operadores de expresiones regulares para Mysql

Name	Description
<u>NOT REGEXP</u>	Negation of REGEXP
<u>REGEXP</u>	Whether string matches regular expression
<u>REGEXP_INSTR()</u>	Starting index of substring matching regular expression
<u>REGEXP_LIKE()</u>	Whether string matches regular expression
<u>REGEXP_REPLACE()</u>	Replace substrings matching regular expression
<u>REGEXP_SUBSTR()</u>	Return substring matching regular expression
<u>RLIKE</u>	Whether string matches regular expression

Fuente: <https://dev.mysql.com/doc/refman/8.4/en/regexp.html>

Expresiones regulares en PDI

Uso REGEX de Mysql

```
SELECT  
  
    idEntidadDocumental  
  
    , lugarPublicacionOriginal  
  
    , reproduccion  
  
FROM MANIFESTACION m  
  
WHERE reproduccion REGEXP '[Vv]o1\.(umen)?';
```

The screenshot shows the 'Table input' dialog box with the following configuration:

- Step name:** getWORKS_vol
- Connection:** LOCAL
- SQL:**

```
SELECT  
    idEntidadDocumental  
    , reproduccion  
FROM MANIFESTACION m  
WHERE reproduccion REGEXP '[Vv]o1\.(umen)?';
```
- Line 3 Column 0** (highlighted)
- Store column info in step meta data:** ☐
- Enable lazy conversion:** ☐
- Replace variables in script?:** ☐
- Insert data from step:**
- Execute for each row?:** ☐
- Limit size:** 0

Buttons at the bottom: ? Help, OK, Preview (highlighted), Cancel.

Expresiones regulares en PDI

Uso REGEX de Mysql

Resultado:

Edició digital basada en l' edició d'Alexandre Micha, Genève, Droz 1978-1983, **9vols.**

Altra ed.: *Catalan Review*, **Volume** VI, numbers 1-2 (1992), pp. 393-399.

Otra ed.: *Celestinesca*, **vol.** 14, núm. 2 (nov. 1990), pp. 3-39

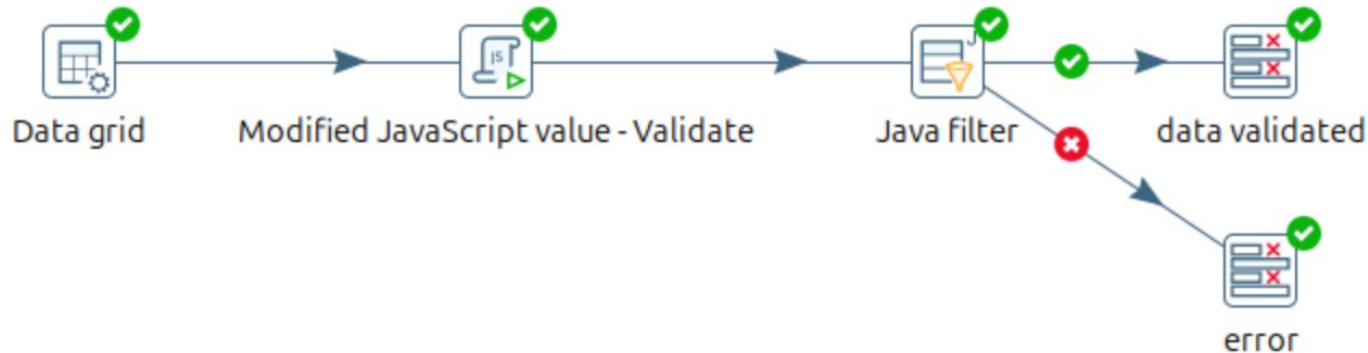
Edición digital a patir de la de Valencia, Herederos de Gerónimo Conejos, 1752, **9vols.**

Altra ed.: *Estudios Lulianos*, **Vol.** 29 (1989), Fasc. 1, pp. 1-23. Fasc. 2, pp. 101-124

Expresiones regulares en PDI

JavaScript

Variante del ejercicio 2 - Validar una contraseña mediante una expresión regular. La contraseña debe tener entre 8 y 16 caracteres y contener al menos una letra minúscula, una mayúscula y un número.



Expresiones regulares en PDI

Modified JavaScript value



Modified JavaScript value

Step name: **Modified JavaScript value - Validate**

JavaScript functions:

- Transform Scripts
- Transform Constant
- Transform Function
- Input fields
 - passTest
- Output fields

Please use the 'Input fields' section to select the fields to be processed.

JavaScript:

```
//Script here

/**
 * Validar contraseña con una expresión regular.
 * @param {string}
 * @returns {boolean} true si es válida y false si no lo es
 */
function isValid(pass) {
    const regex = /^(?=.*[a-z]) (?=.*[A-Z]) (?=.*\d) .{8,16}$/;
    return regex.test(pass);
}

var valid = isValid(passTest);
```

Linernr: 0

Compatibility mode ☐ Optimization level 9

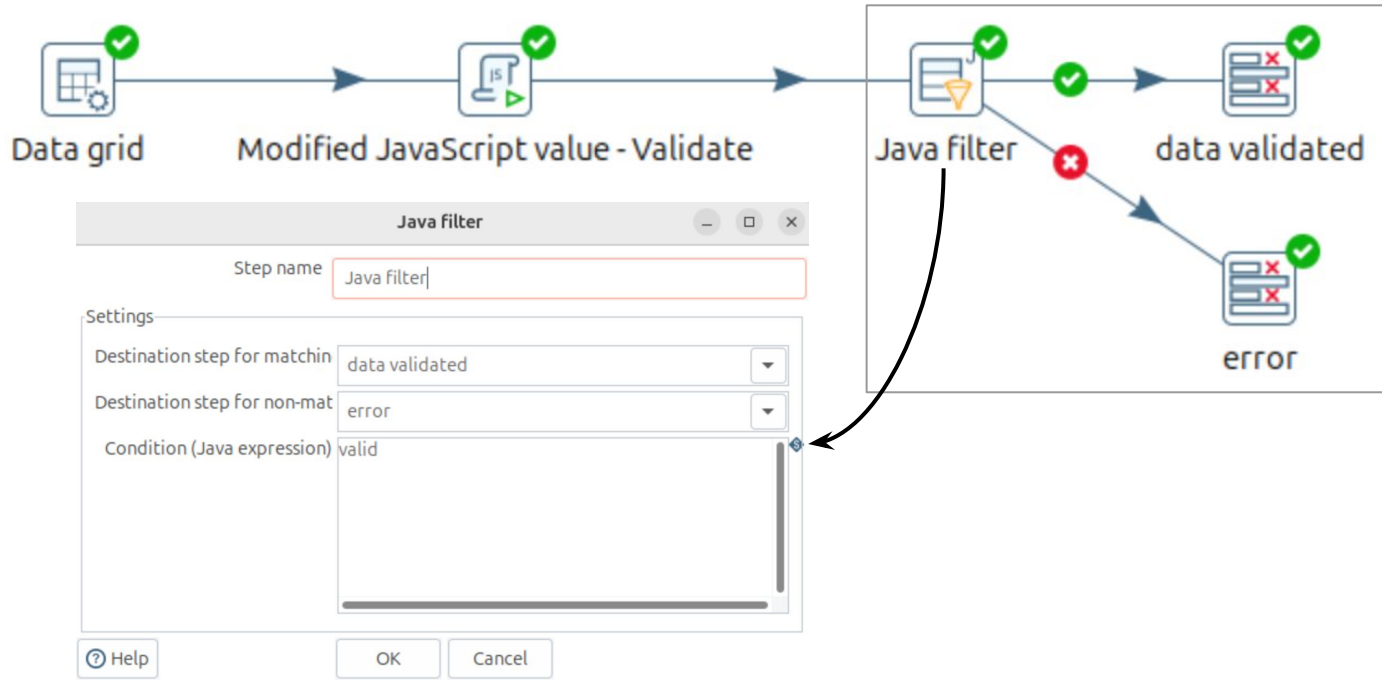
Fields

Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1 valid		Boolean			N

Help OK Cancel Get variables Test script

Expresiones regulares en PDI

Java filter



Referencias

- Kelhini, F. K. (2024). Text processing with JavaScript : regular expressions, tools, and techniques for optimal performance / (First edition.). The Pragmatic Programmers, LLC.
- [MySQL 8.4 Reference Manual](https://dev.mysql.com/doc/refman/8.4/en/regexp.html), Regular Expressions
dev.mysql.com/doc/refman/8.4/en/regexp.html
- [Learn SQL SQL Server](https://learn.microsoft.com/es-es/sql/relational-databases/regular-expressions/overview?view=sql-server-ver17)
learn.microsoft.com/es-es/sql/relational-databases/regular-expressions/overview?view=sql-server-ver17
- Pentaho Data Integration Regex Evaluation
pentaho-public.atlassian.net/wiki/spaces/EAI/pages/371558254/Regex+Evaluation (earlier version)