



INSTITUTO POLITÉCNICO NACIONAL.

ESCUELA SUPERIOR DE CÓMPUTO.

TEORÍA COMPUTACIONAL

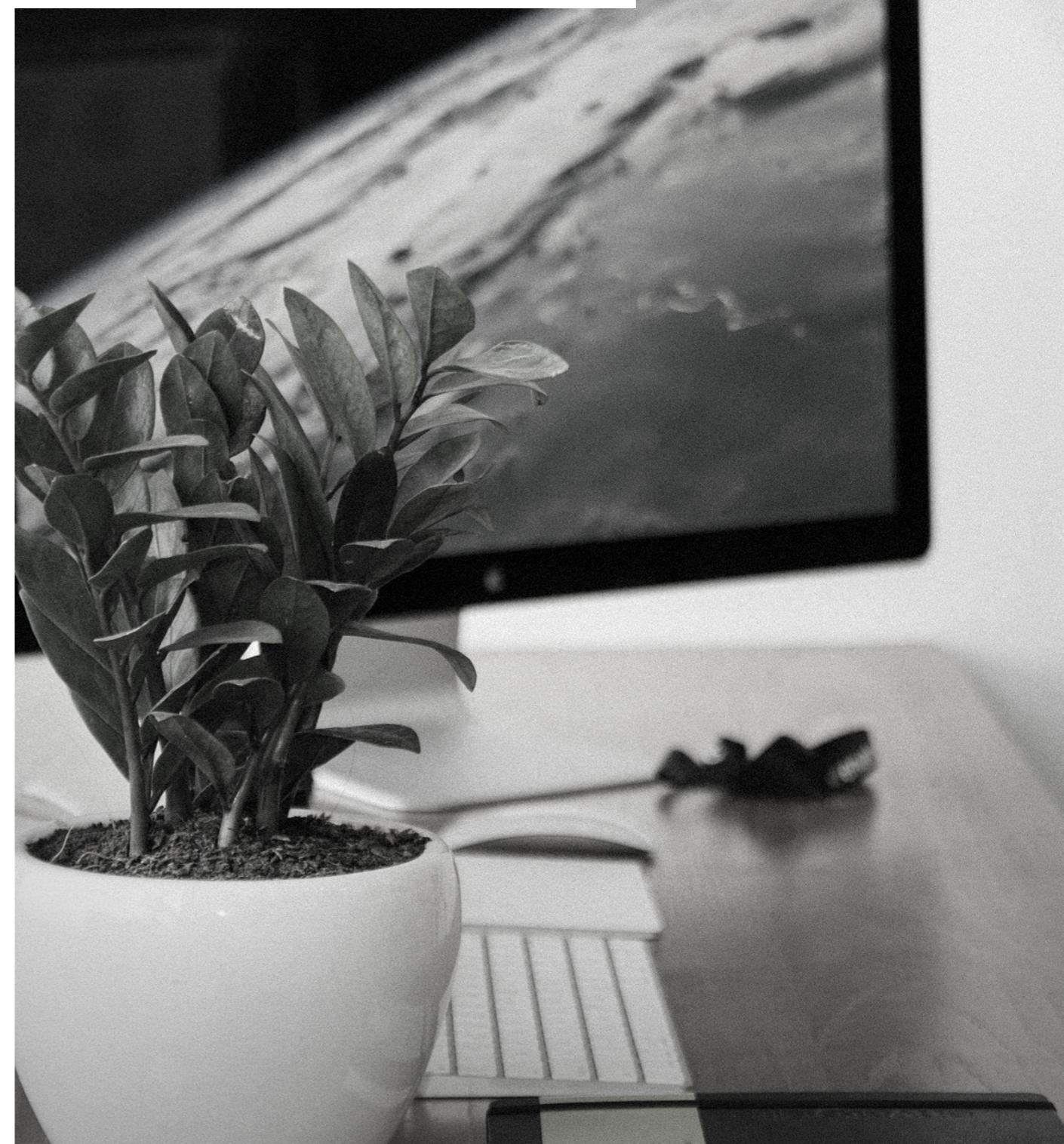
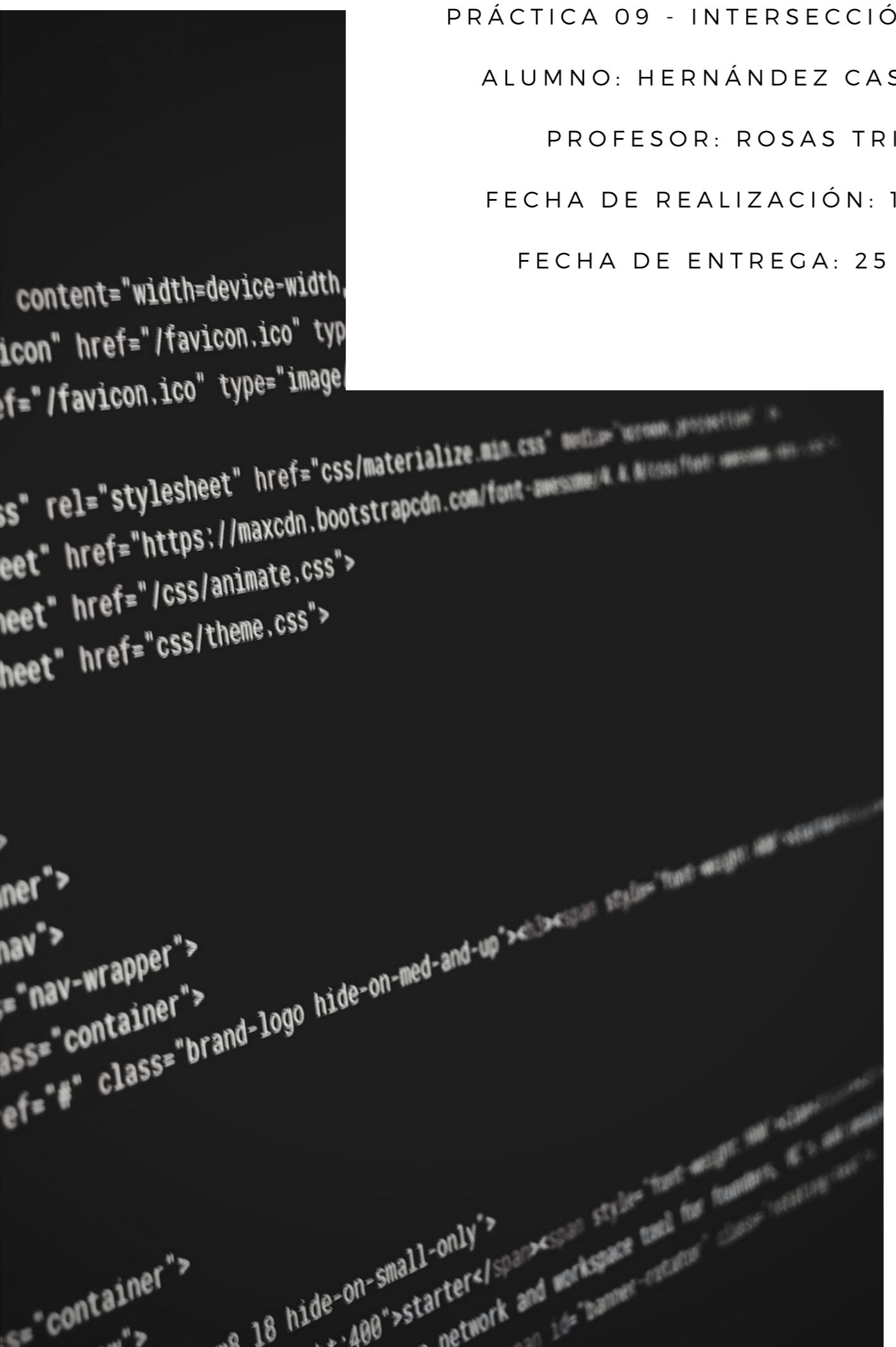
PRÁCTICA 09 - INTERSECCIÓN Y UNIÓN DE LENGUAJES.

ALUMNO: HERNÁNDEZ CASTELLANOS CÉSAR URIEL.

PROFESOR: ROSAS TRIGUEROS JORGE LUIS

FECHA DE REALIZACIÓN: 18 DE OCTUBRE DEL 2017

FECHA DE ENTREGA: 25 DE OCTUBRE DEL 2017



Marco teórico.

¿Qué es un lenguaje?

Un Lenguaje es una subconjunto del Lenguaje Universal ($W(E)$), es decir, es un conjunto de cadenas o palabras obtenidas de un alfabeto en particular.

Al ser el Lenguaje Universal un conjunto infinito, los lenguajes asociados a él también lo son.

Simbología: L

$$\begin{aligned}L_1 (\Sigma_2) &= \{a, ae, eia, oua\} \\L_2 (\Sigma_2) &= \{ee, ae, eia, o, uaei\}\end{aligned}$$

Imagen 1.0 – Lenguajes.

Operaciones con lenguajes.

Unión.

La unión de dos lenguajes (L_1 y L_2) es un nuevo lenguaje formado por el conjunto de todas las palabras que pertenecen tanto al primer lenguaje como al segundo.

Simbología: $L_1 \cup L_2$

Ejemplo:

$$\begin{aligned}L_1 (\Sigma_2) &= \{a, ae, eia\} \\L_2 (\Sigma_2) &= \{ee, ae, eia, o\} \\L_1 \cup L_2 &= \{a, ae, eia, ee, o\}\end{aligned}$$

Imagen 1.1 – Unión de lenguajes.

La unión de dos lenguajes regulares es otro lenguaje regular. Se utiliza la operación de unión de conjuntos; así, para el alfabeto $S = \{x,y\}$ si $L_1 = \{x,xy\}$ y $L_2 = \{yz,yy\}$ entonces su unión será

$$L_1 \cup L_2 = \{x,xy,yz,yy\}.$$

Para modificar los diagramas de transiciones deberemos conseguir que se vaya a una u otra de las estructuras originales, pero sin mezclarlas (si se mezclan, se admitirían cadenas que no pertenecen al lenguaje). Para ello cogeremos los diagramas originales de los dos lenguajes. Se dibuja un estado nuevo:

Será el nuevo estado inicial

Será de aceptación si alguno de los estados iniciales de los diagramas originales lo era

Por cada uno de los arcos que hay desde los estados iniciales originales hacia otros (puede ser el mismo), se dibuja desde el nuevo estado inicial un arco hacia el estado destino del arco correspondiente en el diagrama original y se etiqueta con el mismo símbolo

A continuación se elimina la característica de inicio de los estados iniciales originales.

Intersección.

La Intersección de dos Lenguajes (L_1 y L_2), es un nuevo Lenguaje formado por el conjunto de las palabras que tienen en común ambos lenguajes.

Símbología:

$$L_1 \cap L_2$$

El lenguaje vacío es aquel Lenguaje que no contiene palabras, distinto de aquel Lenguaje que contiene solo la palabra vacía, contiene una palabra.

Ejemplo:

$$\begin{aligned} L_1 (\Sigma^2) &= \{a, ae, eia\} \\ L_2 (\Sigma^2) &= \{ee, ae, eia, o\} \\ L_1 \cap L_2 &= \{ae, eia\} \end{aligned}$$

Imagen 1.2 – Intersección de lenguajes.

La intersección de varios lenguajes regulares es otro lenguaje regular. Se utiliza la operación de intersección de conjuntos; así, para el alfabeto $S = \{x,y\}$ si $L_1 = \{x,xy,yy\}$, $L_2 = \{yz,yy\}$ y $L_3 = \{y,yy\}$ entonces su intersección será $L_1 \cap L_2 \cap L_3 = \{yy\}$.

Para diseñar el autómata que reconoce este lenguaje, vamos a considerar los autómatas finitos $L(M_n)$. El nuevo autómata estará definido por la quintupla $(S', S', \delta', i', F')$, donde:

S' será el producto cartesiano de todos los conjuntos de estados originales $S' = S_1 \times S_2 \times S_3 \times \dots \times S_n$.

El alfabeto tiene que ser el mismo para todos los autómatas. $S' = S$.

La transición desde el estado $p=(p_1,p_2,p_3,\dots,p_n)$ al estado $q=(q_1,q_2,q_3,\dots,q_n)$ de S' (todas las transiciones posibles), para un símbolo del alfabeto, si y solo si existe la transición desde p_i a q_i para todo $i \in n$.

El estado inicial será aquel que está formado por los estados iniciales originales: $i' = (i_1, i_2, i_3, \dots, i_n)$.

Los estados de aceptación serán aquellos que están formados por estados de aceptación originales. $F' = (F_1, F_2, F_3, \dots, F_n)$.

Material y equipo.

El material utilizado en la práctica es el siguiente:

Herramientas de software:

- * Oracle VM VirtualBox
- * Ubuntu 17.04
- * Python 3.6.2
- * eric6 Web Browser (QtWebKit)
- * JFlap

Herramientas de hardware:

- * Computadora personal

Desarrollo de la práctica.

La práctica número nueve de teoría computacional, consistió en implementar un programa informático que contara con la capacidad de realizar la unión e intersección de dos lenguajes dados, para esto el profesor previamente en clase nos impartió la teoría del como realizar dichar unión e intersección, en el laboratorio únicamente se implementó.

En cuanto a la práctica, no hubo ningún problema a la hora de su realización.

A continuación presento el código empleado para dar resolución al problema además de algunos resultados mostrados en consola.

```

#-*- coding: utf-8 -*-
#!/usr/bin/python
import itertools
SIGMA=['1','0']

## 
Q1=['A','B']
Sa1='A'
SaF=['B']
DELTAnulo={('A','0'):'A',
           ('A','1'):'B',
           ('B','1'):'A',
           ('B','0'):'A'}
## 
Q2=['C','D']
Sb1='C'
SbF=['C']
DELTAdos={('C','1'):'C',
           ('C','0'):'D',
           ('D','1'):'C',
           ('D','0'):'D'}

#ESTADOS
Q=list(itertools.product(Q1,Q2))
# ESTADO INICIAL
S=(Sa1,Sb1)
#ESTADOS FINALES
estaFinalesunion=[]
for q in Q:
    if q[0] in SaF and q[1] in SbF:
        estaFinalesunion.append(q)

estaFinalesinterseccion=[]

for q in Q:
    if q[0] in SaF or q[1] in SbF:
        estaFinalesinterseccion.append(q)

delta={}
#DELTA
for m in Q:
    for i in SIGMA:
        delta[m,i] = (DELTAnulo[m[0],i],DELTAdos[m[1],i])
        print("-----")
        print (" "+str(m[0])+" , "+str(i)+" ")
        print (" "+str(m[1])+" , "+str(i)+" ")
        print("-----")

print(Q)
print(S)
print("Interseccion: "+str(estaFinalesunion))
print("Union: "+str(estaFinalesinterseccion))
print (str(delta))

```

Imagen 1.3 – Código empleado.

```

(A , 1)
(C , 1)
-----
(A , 0)
(C , 0)
-----
(A , 1)
(D , 1)
-----
(A , 0)
(D , 0)
-----
(B , 1)
(C , 1)
-----
(B , 0)
(C , 0)
-----
(B , 1)
(D , 1)
-----
(B , 0)
(D , 0)
-----
[('A', 'C'), ('A', 'D'), ('B', 'C'), ('B', 'D')]
('A', 'C')
Intersection: [('B', 'C')]
Union: [('A', 'C'), ('B', 'C'), ('B', 'D')]
((('A', 'C'), '0'): ('A', 'D'), ((('B', 'C'), '1'): ('A', 'C'), ((('B', 'D'), '1'): ('A', 'C'), ((('B', 'D'), '0'): ('A', 'D'), ((('A', 'D'), ('B', 'C'), '0'): ('A', 'D'), ((('A', 'D'), '1'): ('B', 'C'))])
uriel@Uriel-PC:~/Escritorio$ █

```

Imagen 1.4 – Resultados al ejecutar.

Conclusiones y recomendaciones.

Los lenguajes resultan ser una herramienta poderosa para un programador, ya que reduce el tiempo de desarrollo de manera considerable, además de contar con un código más compacto.

Con el conocimiento de las operaciones entre estos lenguajes, se amplia aún mas su capacidad, pudiendo constituir estructuras más complejas.

Referencias

- [1] *Teoría de Lenguajes, Gramáticas y Autómatas Para Informáticos.* [Place of publication not identified]: Digitalia, Inc., 2000.
- [2] Bernard, "Operaciones entre lenguajes", *Es.slideshare.net*, 2017. [Online]. Available: <https://es.slideshare.net/jeanmbernard37/operaciones-entre-lenguajes>. [Accessed: 18- Oct- 2017].
- [3] "Operaciones con lenguaje: Unión,concatenación y potenciación - Teoria de Automatas", *Sites.google.com*, 2017. [Online]. Available: <https://sites.google.com/site/tautomatasfinal/operaciones-con-lenguaje-union-concatenacion-y-potenciacion>. [Accessed: 18- Oct- 2017].
- [4] "Operaciones con lenguajes", *Delta.cs.cinvestav.mx*, 2017. [Online]. Available: [http://delta.cs.cinvestav.mx/~mcintosh/comun/summer2006/algebraPablo_html/node6.html](http://delta.cs.cinvestav.mx/~mcintosh/comun/summer2006/algebraPablo_html/nоде6.html). [Accessed: 18- Oct- 2017].