

Compilad

Análisis Léxico
ores *Rec.*

tokens

Patrones de lexemas

```
instr  →  if expr then instr
        |  if expr then instr else instr
        |  ε
expr   →  term oprel term
        |  term
term   →  id
        |  numero
```

Figura 3.10: Una gramática para las instrucciones de bifurcación

```
digito  →  [0-9]
digitos  →  digito+
numero  →  digitos ( . digitos )? ( E [+-]? digitos )?
letra   →  [A-Za-z]
id       →  letra ( letra | digito )*
if       →  if
then     →  then
else     →  else
oprel    →  < | > | <= | >= | = | <>
```

Figura 3.11: Patrones para los tokens del ejemplo 3.8

```
ws → ( blanco | tab | nuevalinea )+
```

Tokens, patrones y valores

LEXEMAS	NOMBRE DEL TOKEN	VALOR DEL ATRIBUTO
Cualquier <i>ws</i>	–	–
if	if	–
Then	then	–
else	else	–
Cualquier <i>id</i>	id	Apuntador a una entrada en la tabla
Cualquier <i>numero</i>	numero	Apuntador a una entrada en la tabla
<	oprel	LT
<=	oprel	LE
=	oprel	EQ
<>	oprel	NE
>	oprel	GT
>=	oprel	GE

Figura 3.12: Tokens, sus patrones y los valores de los atributos

Autóma tas

Diagramas de transición

Diagrama de transición

- Un diagrama de transición es la representación gráfica de un patrón o expresión regular
- Está formado de nodos o círculos llamados estados
- Cada estado es una condición derivada del lexema
- Cada estado se conecta a otro a través de líneas etiquetadas con símbolos
- Un diagrama posee un estado de inicio y estados de finalización o aceptación.

Diagrama de transición

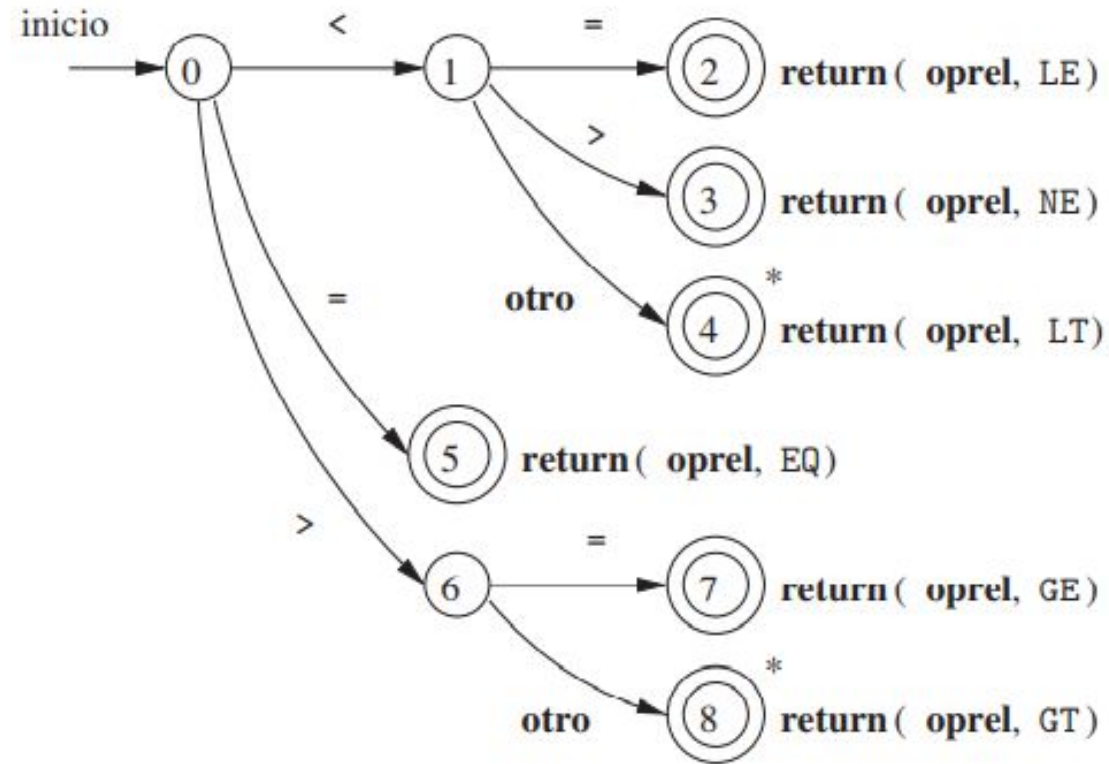


Figura 3.13: Diagrama de transición de estados para **oprel**

identificadores

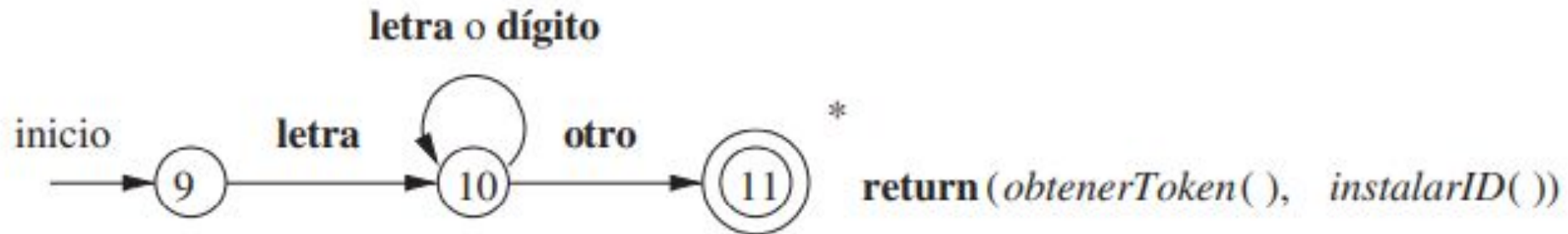


Figura 3.14: Un diagrama de transición de estados para identificadores (**id**) y palabras clave

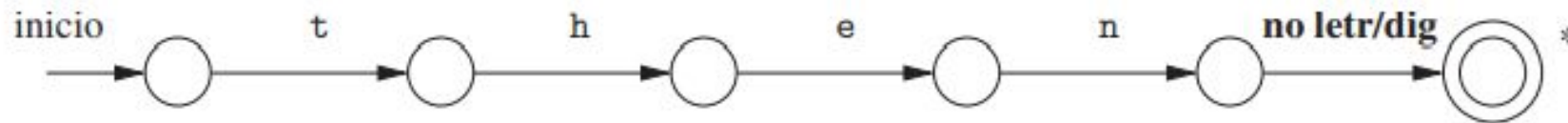


Figura 3.15: Diagrama de transición hipotético para la palabra clave **then**

Números sin signo

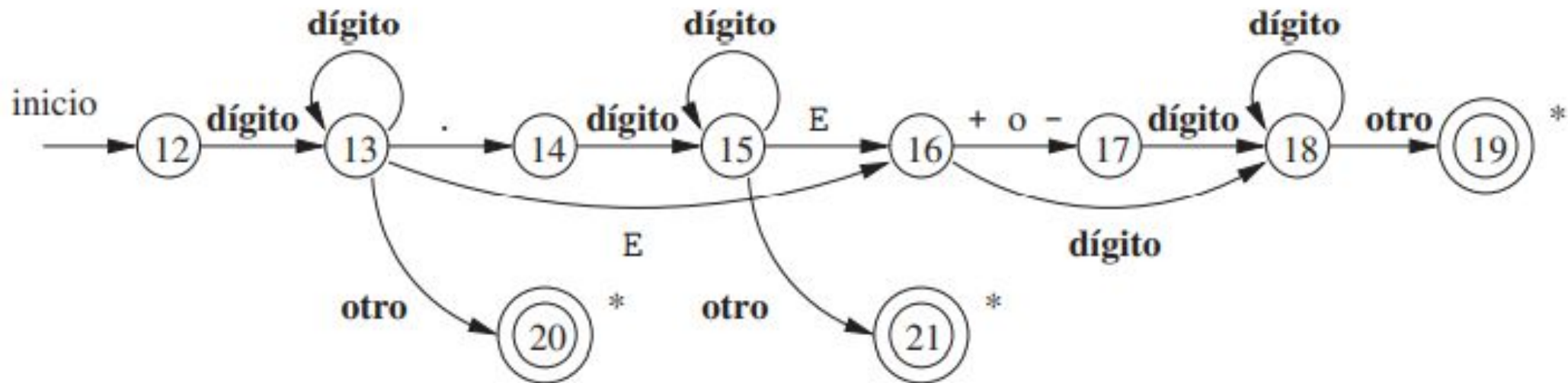


Figura 3.16: Un diagrama de transición para los números sin signo

Delimitador

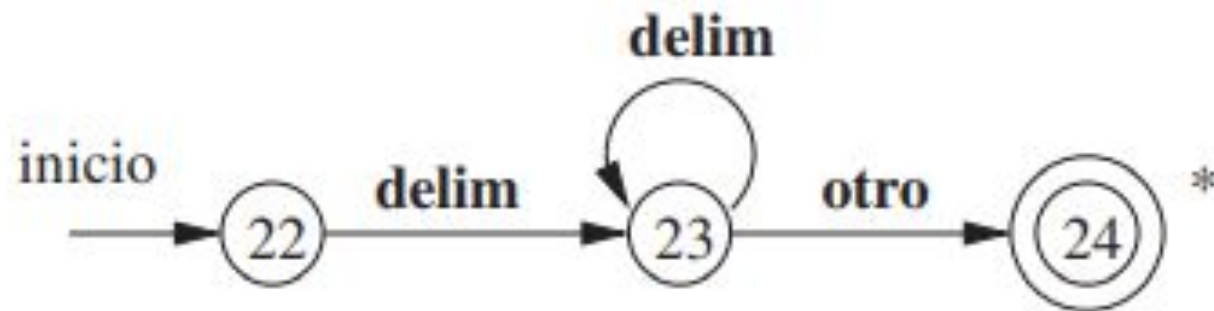


Figura 3.17: Un diagrama de transición para el espacio en blanco



Autómatas Finitos

Autómatas finitos

- Un autómatata finito es un reconocedor de cadenas y pueden de tipo AFN o AFD
- Un AFN
 - No tiene restricciones en las etiquetas, incluye ϵ
 - Son fáciles de generar y difíciles de programar
- Un AFD
 - Limita las etiquetas de salida de los estados a una. No existe ϵ .
 - Son fáciles de programar

AFN

- Posee:
 - Es un conjunto de estados S
 - Un conjunto de símbolos de entrada Σ donde ϵ no pertenece a Σ
 - Una función de transición para cada estado y para cada símbolo $\Sigma \cup \{\epsilon\}$
 - Un estado S_0 de S que es el estado inicial
 - Un conjunto de estados F , suconjunto de S , que representan los estados finales o de aceptación

AFN

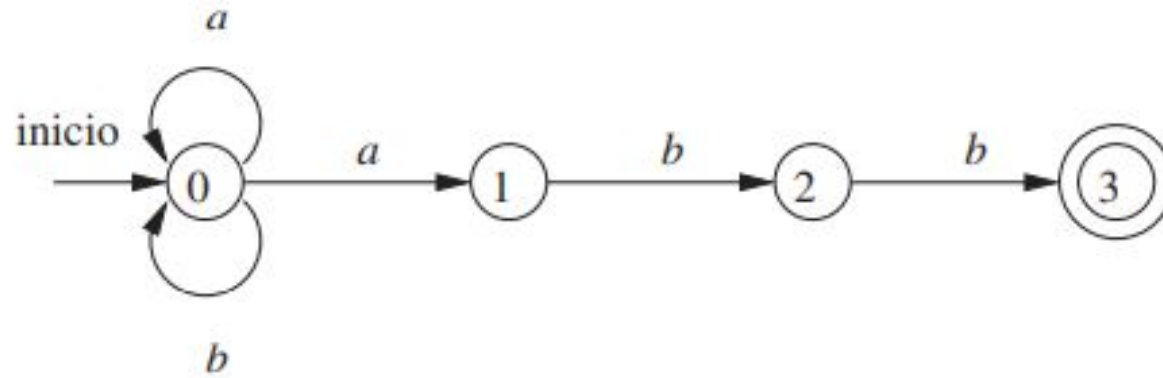


Figura 3.24: Un autómata finito no det

ESTADO	a	b	ϵ
0	$\{0, 1\}$	$\{0\}$	\emptyset
1	\emptyset	$\{2\}$	\emptyset
2	\emptyset	$\{3\}$	\emptyset
3	\emptyset	\emptyset	\emptyset

Figura 3.25: Tabla de transición para el AFN de la figura 3.24

Ejercicios

- Cómo sería el AFN que reco
– $a|(ab)^+$
– $bc|c^*$
– $(a|b)^*c|b$
– $(abb|bba)^*$
– $a^*b^*a^*$
– $aa^*|bb^*$

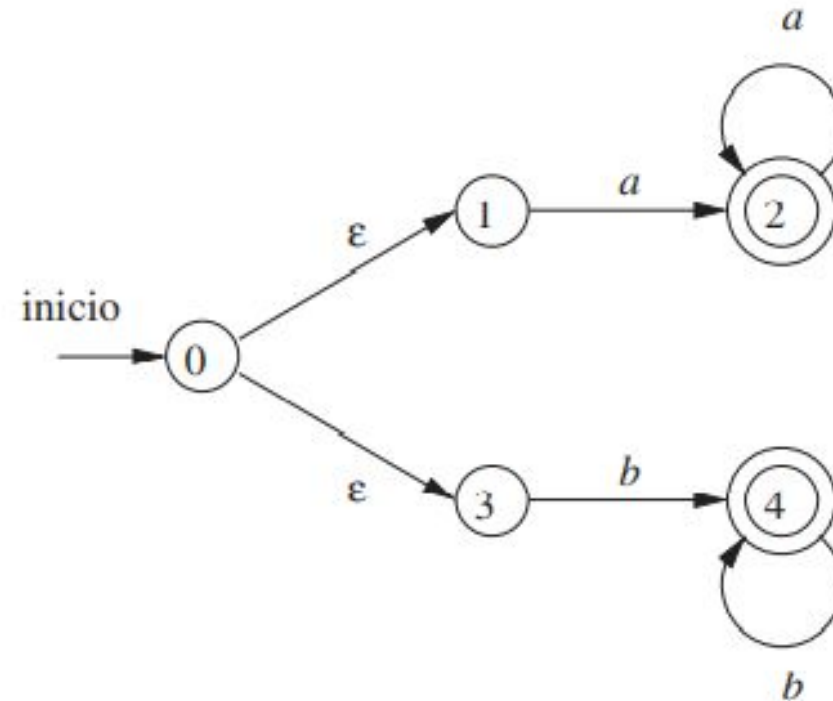


Figura 3.26: AFN que acepta a $aa^*|bb^*$

AFN

- ¿Qué cadenas reconoce?

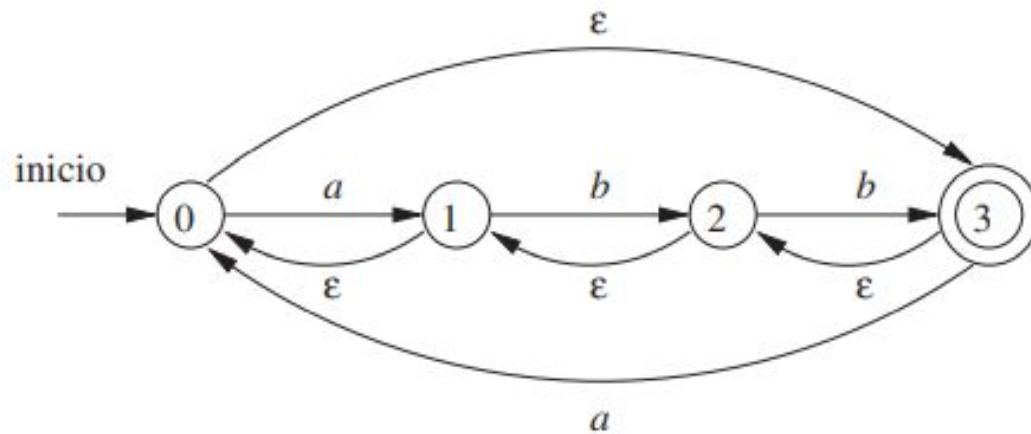


Figura 3.30: AFN para el ejercicio 3.6.4

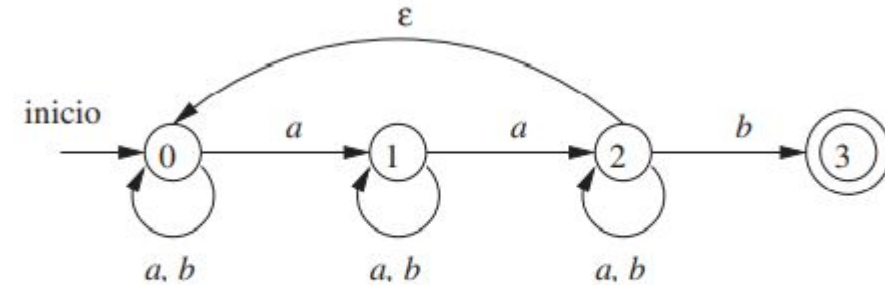


Figura 3.29: AFN para el ejercicio 3.6.3

AFD

- Es un caso especial de AF donde
 - Cada estado tiene un solo destino con la misma etiqueta
 - No hay é

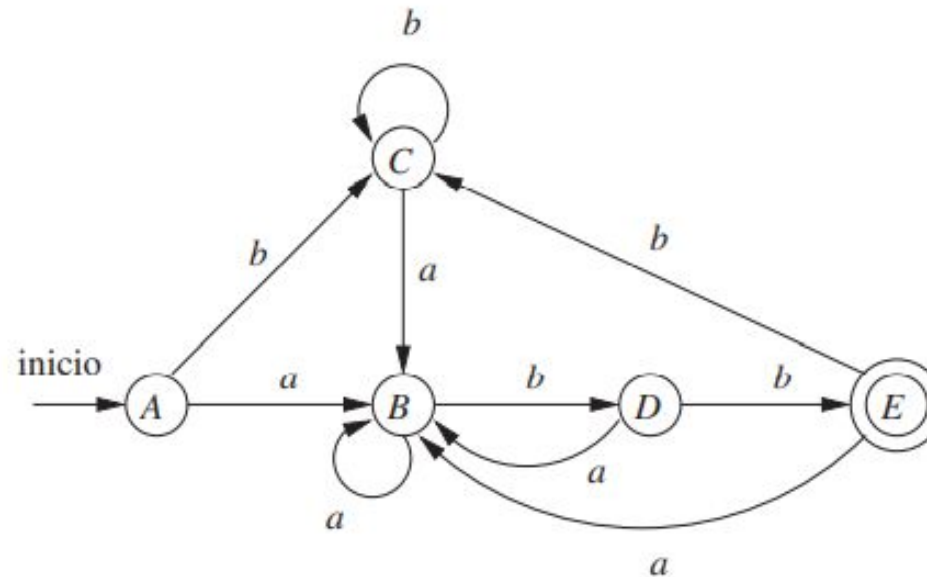


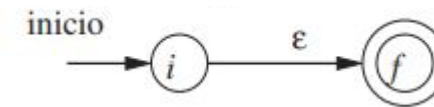
Figura 3.36: Resultado de aplicar la construcción de subconjuntos a la figura 3.34

Programación de un AFD

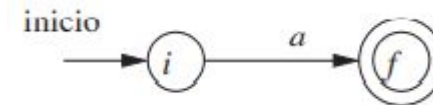
AFN a partir de una exp. regular

- Algoritmo de McNaughton-Yamada-Thompson

- ϵ



- Para todo a en Σ



- Si $N(s)$ y $N(t)$ son autómatas pa
entonces $s|t$

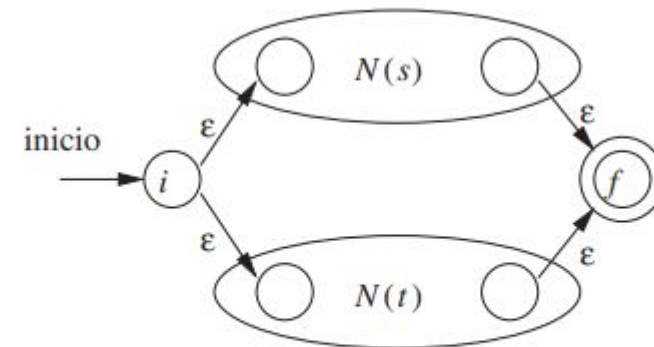


Figura 3.40: AFN para la unión de dos expresiones regulares

AFN a partir de una exp. regular

- Para $r=st$ se tiene
- Para $r=s^*$

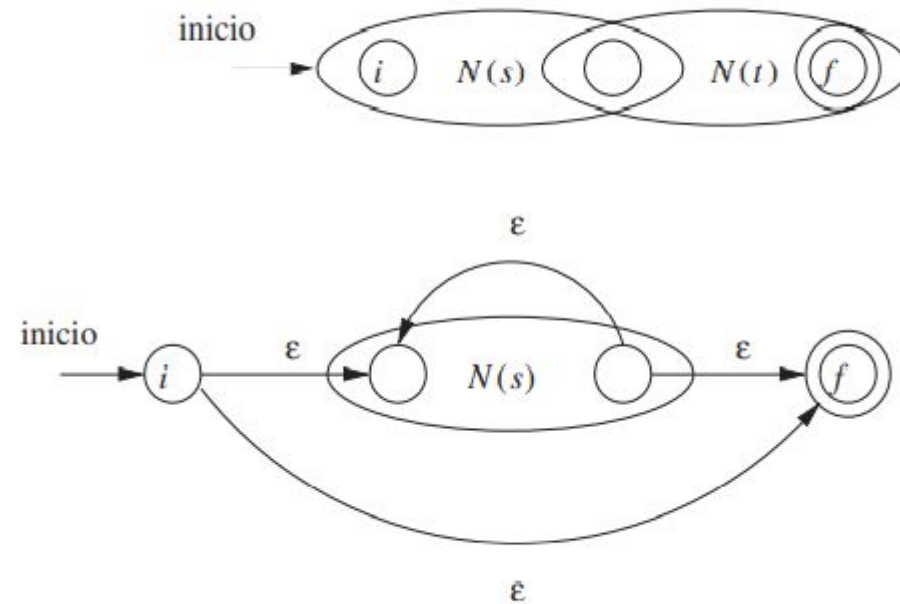


Figura 3.42: AFN para el cierre de una expresión regular

Ejercicio

- Obtenga el AFN- ϵ de la expresión
 - $(a|b)b$
 - $ab^*(a|b)$
 - $(a|b)^*abb$

Ejercicios

- En el diseño de su lenguaje, incluye identificadores, números al menos una sentencia condicional y un ciclo. Elabore AFN que reconozca todos los lexemas de su lenguaje y obtenga los AFD.
- Identificadores
- Números
- Palabras reservadas
- operadores

Referencias

- Aho,
- Imágenes
 - Elaboración propia
 - Aho/Setti Text book

Lexemas y gramáticas

Lexemas



Tokens

- Persona come platillo

- La **gramática** sería

Sentencia ☐ Persona Accion Platillo

Persona ☐ Hugo | Paco | Luis

Platillo ☐ manzana | lechuga | jitomate

Accion ☐ come | elige