

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE LÁZARO CÁRDENAS

LENGUAJES Y AUTÓMATAS I

**“TABLA DE SÍMBOLOS, PALABRAS RESERVADAS Y GRAMÁTICA:
PREVEBOT”**

CATEDRÁTICA:

Araceli Galván Montelongo

ALUMNOS:

Owen Hassan López González

Luis Fernando Rojas González

TABLA DE SÍMBOLOS

Símbolo	Sintaxis	Significado
bla	β	Espacio en blanco
en	\downarrow	Enter
num	#	Es para asignar algún valor numérico
pos	<cadena>	Cadena de caracteres
pra	(Paréntesis que abre
prc)	Paréntesis que cierra
pc	;	Punto y coma
pt	.	Punto
ki	{	Llave que abre
kf	}	Llave que cierra

TABLA DE PALABRAS RESERVADAS

Símbolo	Palabra reservada	Sintaxis	Significado
SP	startPrevebot	$\backslash t^* \beta^* \leftarrow^* \text{startPrevebot} \backslash t^* \beta^* \leftarrow^* \{ \backslash t^* \beta^* \leftarrow^* \}$	Inicio del código
EP	endPrevebot	$\backslash t^* \beta^* \leftarrow^* \text{endPrevebot}; \backslash t^* \beta^* \leftarrow^*$	Fin del código
TN	turn	$\backslash t^* \beta^* \leftarrow^* \text{turn}$	Indica que se hará un giro
RG	right	$\text{.right}(\#); \backslash t^* \beta^* \leftarrow^*$	Debe ir después de un turn, Indica que la dirección de giro será a la derecha
LF	left	$\text{.left}(\#); \backslash t^* \beta^* \leftarrow^*$	Debe ir después de un turn, Indica que la dirección de giro será a la izquierda
ST	straight	$\backslash t^* \beta^* \leftarrow^* \text{straight}(\#);$	Avanza recto
SO	stop	$\backslash t^* \beta^* \leftarrow^* \text{stop}(); \backslash t^* \beta^* \leftarrow^*$	El robot se detiene
WT	wait	$\backslash t^* \beta^* \leftarrow^* \text{wait}(); \backslash t^* \beta^* \leftarrow^*$	Se pone en espera
BC	back	$\backslash t^* \beta^* \leftarrow^* \text{back}(\#); \backslash t^* \beta^* \leftarrow^*$	Retrocede
DT	detect	$\backslash t^* \beta^* \leftarrow^* \text{detect}$	Detecta alguna acción
DR	Door	$\text{.Door}(); \backslash t^* \beta^* \leftarrow^*$	Debe ir después de un “detect” para especificar que se detectará el estado de las puertas

BT	Battery	.Battery();\t*\beta*\↵*	Debe ir después de un “detect”. Detecta el estado de la batería, puede elegir entre correcto, medio y bajo
CS	Colision	.colision();\t*\beta*\↵*	Debe ir después de un “detect” para especificar que se detectará si se ha producido una colisión
OD	openDoor	\t*\beta*\↵*openDoor(0/1);\t*\beta*\↵*	Abre alguna de las puertas: 0 = puerta de carga 1 = Puerta de cargador
CD	closeDoor	\t*\beta*\↵*closeDoor(0/1);\t*\beta*\↵*	Cierra alguna de las puertas: 0 = puerta de carga 1 = Puerta de cargador
TP	transport	\t*\beta*\↵*transport	Le ordena al robot transportar algo
MED	medicine	.medicine(<cadena>);\t*\beta*\↵*	Debe ir después de un “transport” para especificar que se transportarán medicinas
CLO	cloth	.cloth(<cadena>);\t*\beta*\↵*	Debe ir después de un “transport” para especificar que se transportará ropa

SP	sample	.sample(<cadena>);\t*\b*\d*	Debe ir después de un “transport” para especificar que se transportarán muestras
AL	alert	\t*\b*\d*alert	Prepara un mensaje de alerta/aviso
CE	clean	.clean();\t*\b*\d*	Indica que debe ser limpiado y desinfectado
LB	lowBattery	.lowBattery();\t*\b*\d*	Debe ir después de un “alert”, sirve para alertar que la batería está baja

REGLAS GRAMATICALES

Sintaxis	Regla gramatical
\t*\beta*\epsilon*startPrevebot\t*\beta*\epsilon*\{\t*\beta*\epsilon*	ST → BSP Bi B
\t*\beta*\epsilon*endPrevebot;\t*\beta*\epsilon*	FN → BE Pc B INI → STIS IS → B1B2A1 B1B2A1IS TO λ B1 → MOP1 OP1 → TO ODDOC TO → ZFN OC → T V W TOC VOC WOC DD → RL RLDD B2 → OP2 OP2B2 λ OP2 → MOP3 MOP3OP2 OP3 → TO MVCL MV → C E F H CL → N NGCL A1 → OR
\t*\beta*\epsilon*turn	A → BT N B → TAB BLA EN TAB → tab tab TAB λ BLA → bla bla BLA λ EN → en en EN λ
.right(#);\t*\beta*\epsilon*	C → Apt RG DB D → pranumprepc
.left(#);\t*\beta*\epsilon*	E → Apt LF DB
\t*\beta*\epsilon*straight(#);	F → BST D

<code>\t*\beta*\epsilon*stop();\t*\beta*\epsilon*</code>	G → BSPHB H → praprcpc
<code>\t*\beta*\epsilon*wait();\t*\beta*\epsilon*</code>	I → BWTHB
<code>\t*\beta*\epsilon*back(#);\t*\beta*\epsilon*</code>	J → BBCDB
<code>\t*\beta*\epsilon*detect</code>	K → BDT
<code>.Door();\t*\beta*\epsilon*</code>	L → Kpt DRHB
<code>.Battery();\t*\beta*\epsilon*</code>	M → Kpt BTHB
<code>.colision();\t*\beta*\epsilon*</code>	N → Kpt CSHB
<code>\t*\beta*\epsilon*openDoor(0/1);\t*\beta*\epsilon*</code>	O → BODPB P → praQprcpc Q → 0 1
<code>\t*\beta*\epsilon*closeDoor(0/1);\t*\beta*\epsilon*</code>	R → BCDPB
<code>\t*\beta*\epsilon*transport</code>	S → BTP
<code>.medicine(<cadena>);\t*\beta*\epsilon*</code>	T → Spt MEDUB U → praposprcpc
<code>.cloth(<cadena>);\t*\beta*\epsilon*</code>	V → Spt CLOUB
<code>.sample(<cadena>);\t*\beta*\epsilon*</code>	W → Spt SPUB
<code>\t*\beta*\epsilon*alert</code>	X → BALT
<code>.clean();\t*\beta*\epsilon*</code>	Y → Xpt CLHB
<code>.lowBattery();\t*\beta*\epsilon*</code>	Z → Xpt LBHB