Gramática de **OBLIQ**

Escuela de Ingeniería de Sistemas y Computación

Facultad de Ingeniería

	::=	<bool-expresion>
⟨programa⟩	::=	⟨expresión⟩
⟨expresión⟩	::=	⟨identificador⟩
	::=	⟨numero⟩
	::=	⟨caracter⟩
	::=	⟨cadena⟩
	::=	ok
	::=	var { ⟨identificador⟩ = ⟨expresión⟩}$^{*(,)}$ in ⟨expresión⟩ end
	::=	let { ⟨identificador⟩ = ⟨expresión⟩}$^{*(,)}$ in ⟨expresión⟩ end
	::=	let rec { ⟨identificador⟩ ({⟨identificador⟩}$^{*(,)}$) = ⟨expresión⟩}* in ⟨expresión⟩ end
	::=	set ⟨identificador⟩ := ⟨expresión⟩
	::=	begin <expresion>; {<expresion>}*(;)
	::=	⟨primitiva⟩ ({⟨expresión⟩}*)
	::=	if ⟨bool-expresión⟩ then ⟨expresión⟩ {elseif ⟨bool-expresión⟩ then ⟨expresión⟩ }* else ⟨expresión⟩ end
	::=	proc ({⟨identificador⟩}$^{*(,)}$) ⟨expresión⟩ end
	::=	apply ⟨identificador⟩ ({⟨expresión⟩}$^{*(,)}$)
	::=	meth (⟨identificador⟩, {⟨identificador⟩}$^{*(,)}$) ⟨expresión⟩ end
	::=	for ⟨identificador⟩ = ⟨expresión⟩ to ⟨expresión⟩ do ⟨expresión⟩ end
	::=	object { ⟨identificador⟩ => ⟨expresión⟩ }* end
	::=	get ⟨identificador⟩.⟨identificador⟩
	::=	send ⟨identificador⟩.⟨identificador⟩({⟨identificador⟩}$^{*(,)}$)
	::=	update ⟨identificador⟩.⟨identificador⟩ := ⟨expresión⟩
	::=	clone (⟨identificador⟩)
⟨bool-expresión⟩	::=	⟨bool-primitiva⟩({⟨expresión⟩}*)
	::=	⟨bool-oper⟩({⟨bool-expresión⟩}*)
	::=	true
	::=	false
⟨primitiva⟩	::=	$+ \mid - \mid * \mid / \mid \% \mid \&$
⟨bool-primitiva⟩	::=	$< \mid > \mid <= \mid >= \mid is$
⟨bool-oper⟩	::=	$not \mid and \mid or$