

ANÁLISIS SINTÁCTICO

>> Análisis del lenguaje <<

No.	Regla de Producción	Estructura(s) que analiza	Objetivo	Error reconocido
1	INICIO -> IMPORTS CLASE CLASE	Estructura general de una clase Java	Permitir que se pueda empezar a redactar una clase Java	-
2	IMPORTS -> import IIMPORT IIMPORT	Un bloque declaración de imports, conformado por 1 o más líneas de imports	Permitir que pueda declararse imports	X
3	IIMPORT -> import DIRECCION_IMPORT;	Una línea de imports	Permitir declarar un import	-
4	IMPORT -> DIRECCION , * DIRECCION	Una sentencia completa o parcial de import	Permitir que sea importado uno o más elementos de la dirección especificada	X
5	DIRECCION -> DIRECCION , nombre nombre , nombre	El path a importar	Permitir el listado para crear el path a importar	-
6	CLASE -> VISIBILIDAD DEF_CLASE DEF_CLASE	La visibilidad de la firma de una clase	Permitir que el usuario pueda declarar visibilidad	X
7	VISIBILIDAD -> public protected . private final	Los tipos de visibilidad que son aceptados en Java	Permitir que se definan los diferente tipos de visibilidad que el usuario pueda definir	-
8				
9	DEF_CLASE -> class objeto { BLOQUE_CLASE } class objeto { }	La parte restante de la firma de una clase	Permitir que un usuario pueda declara la estructura de una clase	-
10	BLOQUE_CLASE -> BLOQUE_CLASE SENTENCIAS_CLASE SENTENCIAS_CLASE	El bloque de sentencias aceptado en una clase	Permitir que un usuario pueda definir el bloque de una clase	-
11				
12	SENTENCIAS_CLASE -> DEF_VARIABLE DEF_METODO	El listado de sentencias de una clase que pueden aparecer swnreo de ella	Listar los difernetes tipos de sentencias que pueden econtrarse en las clases	-
13				
14	DEF_VARIABLE -> VISIBILIDAD DEF_TIPO_VARIABLE; DEF_TIPO_VARIABLE;	La visibilidad y la estructura de la definición de una variable	Permitir que una variable pueda tener visibilidad	-
15	DEF_TIPO_VARIABLE -> int CREACION_VARIABLE:_NUMERO double CREACION_VARIABLE:_NUMERO string CREACION_VARIABLE_STRING char CREACION_VARIABLE_CHAR boolean CREACION_VARIABLE_BOOLEAN object CREACION_VARIABLE_OBJECT objeto CREACION_VARIABLE_OBJETO	Los tipos de variables que pueden ser definidas	Listar los difernetes tipos de variable que puede ser creada	X
16	CREACION_VARIABLE_NUMERO -> CREACION_VARIABLE_NUMERO , VARIABLE_NUMERO VARIABLE_NUMERO	La sintaxis general para la creación de variables numéricas	Permitir la creaciión de una o más variables	X
17	VARIABLE_NUMERO -> nombre = CONTENIDO_NUMERO nombre = CONTENIDO_CHAR nombre	Los diferentes tipos de forma en las que se puede crear una variable de tipo numérico en el lenguaje Java	Permitir la definición de una sola variable a crear	-
18	CREACION_VARIABLE_STRING -> CREACION_VARIABLE_STRING , VARIABLE_STRING VARIABLE_STRING	La sintaxis general para la creación de varibales de tipo string	Permitir la creaciión de una o más variables de tipo String	X
19	VARIABLE_STRING -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_STRING nombre	Los diferentes tipos de forma en las que se puede crear una variable de tipo numérico en el lenguaje Java	Permitir la definición de una sola variable a crear de tipo String	-
20	CREACION_VARIABLE_CHAR -> CREACION_VARIABLE_CHAR , VARIABLE_CHAR VARIABLE_CHAR	La sintaxis general para la creación de varibales de tipo char	Permitir la creaciión de una o más variables de tipo char	X
21	VARIABLE_CHAR -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_CHAR nombre	Los diferentes tipos de forma en las que se puede crear una variable de tipo char en el lenguaje Java	Permitir la definición de una sola variable a crear de tipo char	-
22	CREACION_VARIABLE_BOOLEAN -> CREACION_VARIABLE_BOOLEAN , VARIABLE_BOOLEAN VARIABLE_BOOLEAN	La sintaxis general para la creación de varibales de tipo boolean	Permitir la creaciión de una o más variables de tipo boolean	X
23	VARIABLE_BOOLEAN -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_BOOLEAN nombre	Los diferentes tipos de forma en las que se puede crear una variable de tipo boolean en el lenguaje Java	Permitir la definición de una sola variable a crear de tipo boolean	-

			24	CREACION_VARIABLE_OBJECT -> CREACION_VARIABLE_OBJECT , VARIABLE_OBJECT VARIABLE_OBJECT	La sintaxis general para la creación de variables de tipo Object	Permitir la creación de una o más variables de tipo Object	X
			25	VARIABLE_OBJECT -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_OBJECT nombre = CONTENIDO_OBJETO nombre	Los diferentes tipos de forma en las que se puede crear una variable de tipo Object en el lenguaje Java	Permitir la definición de una sola variable a crear de tipo Object	-
			26	CREACION_VARIABLE_OBJETO -> CREACION_VARIABLE_OBJECT , VARIABLE_OBJETO VARIABLE_OBJETO	La sintaxis general para la creación de variables de tipo Objeto	Permitir la creación de una o más variables de tipo Objeto personalizado	X
			27	VARIABLE_OBJETO -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_OBJETO nombre	Los diferentes tipos de forma en las que se puede crear una variable de tipo Objeto personalizado en el lenguaje Java	Permitir la definición de una sola variable a crear de tipo Objeto personalizado	-
			28	DEF_METODO -> VISIBILIDAD CUERPO_DEF_METODO CUERPO_DEF_METODO	La estructura de la declaración de un método más el tipo de visibilidad que éste pueda tener	permitir que un método pueda tener una visibilidad asignada	X
			29	CUERPO_DEF_METODO -> DEF_METODO_CONSTRUCTOR DEF_METODO_TIPO DEF_METODO_VOID	Los diferentes tipos de método que pueden ser definidos en una parte reducida del lenguaje Java	listar los diferentes tipos de métodos que puede definir un usuario	-
			30	DEF_METODO_CONSTRUCTOR -> objeto (PARAMETROS) CUERPO_METODO_CONSTRUCTOR	Estructura general de un constructor	Definir el cuerpo de un constructor	X
			31	CUERPO_METODO_CONSTRUCTOR -> { BLOQUE } { }	El cuerpo que puede tener un método de tipo constructor	Definir el tipo de cuerpo que puede contener un constructor	
			32	DEF_METODO_TIPO -> int nombre (PARAMETROS) CUERPO_METODO_NUMERICO double nombre (PARAMETROS) CUERPO_METODO_NUMERICO string nombre (PARAMETROS) CUERPO_METODO_STRING char nombre (PARAMETROS) CUERPO_METODO_CHAR boolean nombre (PARAMETROS) CUERPO_METODO_NUMERICO object nombre (PARAMETROS) CUERPO_METODO_NUMERICO objeto nombre (PARAMETROS) CUERPO_METODO_NUMERICO	Los diferentes tipos de retorno que puede tener un método con un tipo asignado diferente de "void"	Definir la estructura general de un método con tipo de retorno diferente de tipo	-
			33	CUERPO_METODO_NUMERICO -> { BLOQUE RETORNO_NUMERICO } { RETORNO_NUMERO } { BLOQUE RETONO_CHAR } { RETONO_CHAR }	El cuerpo que un método de tipo numérico puede tener	El tipo de bloque que un método de tipo numérico puede poseer	X
			34	CUERPO_METODO_STRING -> { BLOQUE RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE RETONO_STRING } { RETONO_STRING }	El cuerpo que un método de tipo String puede tener	El tipo de bloque que un método de tipo String puede poseer	-
			35	CUERPO_METODO_CHAR -> { BLOQUE RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE RETONO_CHAR } { RETONO_CHAR }	El cuerpo que un método de tipo char puede tener	El tipo de bloque que un método de tipo char puede poseer	X
			36	CUERPO_METODO_BOOLEAN -> { BLOQUE RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE RETONO_BOOLEAN } { RETONO_BOOLEAN }	El cuerpo que un método de tipo boolean puede tener	El tipo de bloque que un método de tipo boolean puede poseer	X
			37	CUERPO_METODO_OBJECT -> { BLOQUE RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE RETONO_OBJECT } { RETONO_OBJECT }	El cuerpo que un método de tipo Object puede tener	El tipo de bloque que un método de tipo Object puede tener	X
			38	CUERPO_METODO_OBJETO -> { BLOQUE RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE RETONO_OBJETO } { RETONO_OBJETO }	El cuerpo que un método de tipo Objeto personalizado puede tener	El tipo de bloque que un método de Objeto personalizado puede tener	X
			39	DEF_METODO_VOID -> void nombre CUERPO_METODO_VOID	La estructura de la firma de un método de tipo void más la visibilidad que éste puede poseer	La firma que un método de tipo void	X
			40	CUERPO_METODO_VOID -> (PARAMETROS) { BLOQUE RETURN ; } (PARAMETROS) { BLOQUE } (PARAMETROS) { RETURN ; } (PARAMETROS) { }	El cuerpo que un método de tipo de retorno vacío (void) puede tener	El cuerpo que un método de tipo void puede poseer dentro de sí	-
			41	PARAMETROS -> PARAMETRO , PARAMETRO PARAMETRO λ	El listado de parámetros a agregar a un método	Permitir que un usuario pueda enviar ninguno, 1 o más parámetros a un método para que se pueda trabajar con el contenido enviado	X
			42	PARAMETRO -> TIPO nombre	La sintaxis que debe poseer un parámetro para ser reconocido como tal	Definir la forma general que un parámetro debe poseer al momento de ser declarado en la forma de un método	X
			43	BLOQUE -> BLOQUE SENTENCIAS SENTENCIAS	El bloque que una estructura: método estructura condicional ciclo, puede contener	Que un usuario pueda declarar una o más sentencias dentro de un bloque de una estructura	-

				44	SENTENCIAS -> DEFINICION_VAR ASIGNACION CICLO INVOCACION SENTENCIA_CONTROL OPERACION_UNARIA;	Las diferentes sentencias que pueden estar definidas dentro de un bloque de un método, estructura condicional o ciclo	Definir los tipos de sentencias que puede poseer las estructuras dentro de sí	-
				45	ASIGNACION -> CUERPO_INVOCACION_VAR = TIPOS_ASIGNACION; this . CUERPO_INVOCACION_VAR = TIPOS_ASIGNACION;	La sintaxis general que una asignación a variable debe poseer	La forma que puede poseer una sentencia de tipo asignación	-
				46	TIPOS_ASIGNACION -> CONTENIDO_NUMERO CONTENIDO_STRING CONTENIDO_CHAR CONTENIDO_BOOLEAN CONTENIDO_OBJECT CONTENIDO_OBJETO this . CUERPO_INVOCACION_VAR CUERPO_INVOCACION_METODO this . CUERPO_INVOCACION_METODO	Los diferentes tipos de valor soportados por una gramática reducida de Java	Que un usuario pueda definir diferentes tipos de datos a un parámetro	-
				47	CONTENIDO_VARIABLE -> (CONTENIDO_VARIABLE) nombre	La forma de colocar o representar un contenido de tipo variable	Que un usuario pueda agrupar o dejar sin paréntesis una variable	-
				48	CONTENIDO_NUMERO -> CONTENIDO_NUMERO + CONTENIDO_NUMERO CONTENIDO_NUMERO - CONTENIDO_NUMERO CONTENIDO_NUMERO * CONTENIDO_NUMERO CONTENIDO_NUMERO / CONTENIDO_NUMERO - CONTENIDO_NUMERO (CONTENIDO_NUMERO) CONTENIDO_NUMERO NOMBRE	Las diferentes operaciones y acciones que definen un valor numérico	Que el usuario por medio de operaciones aritméticas pueda cambiar el valor de un valor entero	-
				49	CONTENIDO_STRING -> CONTENIDO_BOOLEAN + CUALQUIER_CONTENIDO PERMITIDOS	Las diferentes formas, acciones y tipos de contenido que definen un String	Permitir que un usuario pueda crear un valor de tipo String de diferentes formas	X
				50	PERMITIDOS -> CONTENIDO_NUMERO + PERMITIDOS CONTENIDO_CHAR + PERMITIDOS CUALQUIER_CONTENIDO	Las diferentes formas de concatenación que el lenguaje Java admite como válidos	Permitir que el usuario pueda por medio de las reglas léxicas no romper los estándares que Java posee para definir una concatenación	-
				51	CUALQUIER_CONTENIDO -> + CONTENIDO_NUMERO + CONTENIDO_CHAR + CONTENIDO_BOOLEAN + STRINGS	Los diferentes tipos de contenido que son aceptados en una concatenación, con el orden que dicta las reglas de Java	Que el usuario pueda definir los tipos restantes en una concatenación, siempre y cuando cumpla con las reglas	-
				52	STRINGS -> (STRINGS) CADENA	Valores de tipo String, con agrupación infinita permitida	Permitir que un usuario pueda declarar un string ya sea que esté o no agrupado	-
				53	CONTENIDO_CHAR -> (CONTENIDO_CHAR) CARACTER	Valores de tipo String, con agrupación infinita permitida	Permitir que un usuario pueda tener un contenido de tipo char, agrupado o no	-
				54	CONTENIDO_OBJECT -> (CONTENIDO_OBJECT) new OBJECT ()	Valores de tipo Object, con agrupación infinita permitida	Permitir que un usuario pueda tener un contenido de tipo Object, agrupado o no	-
				55	CONTENIDO_OBJETO -> (CONTENIDO_OBJETO) new OBJETO ()	Contenido de un tipo de objeto personalizado, con agrupación infinita permitida	Permitir que un usuario pueda tener un contenido de tipo Objeto personalizado, agrupado o no	-
				56	OPERACION_UNARIA -> INCREMENTO AUTOINCREMENTO	los tipos de operaciones unarias permitidos para enteros y strings	Permitir que un usuario pueda realizar operaciones unarias con los valores numéricos	-
				57	INCREMENTO -> nombre ++ nombre -- ++ nombre -- nombre	Las diferentes formas para realizar un incremento positivo o negativo	Permitir que el usuario solo pueda declarar los tipos de incremento en su código	-
				58	AUTOINCREMENTO -> nombre OPCIONES_AUTOINCREMENTO	La estructura autoincremental que puede poseer una variable String o numérica	Permitir que el usuario pueda definir las opciones de autoincremento	-
				59	OPCIONES_AUTOINCREMENTO -> OPCIONES_AUTOINCREMENTAS - = CONTENIDO_NUMERO * = CONTENIDO_NUMERO / = CONTENIDO_NUMERO	Las diferentes formas de autoincremento	Permitir que el usuario solo pueda utilizar las opciones de autoincremento definidas	-
				60	OPCIONES_AUTOINCREMENTAS -> + = CONTENIDO_NUMERO + = CONTENIDO_STRING	La estructura de autoincremento con la suma, para variables de tipo entero y string	Permitir que los string también puedan ser autoincrementados	-
				63	CICLO -> CICLO_FOR while (CONTENIDO_BOOLEAN) CUERPO_ESTRUCTURAS while (CONTENIDO_VARIABLE) CUERPO_ESTRUCTURAS do CUERPO_ESTRUCTURAS while (CONTENIDO_BOOLEAN); do CUERPO_ESTRUCTURAS while (CONTENIDO_VARIABLE);	Los diferentes ciclos de Java	Permitir que el usuario pueda declarar los tipos existentes en la gramática de Java	-

			64	CICLO_FOR -> for (ASIGNACION_FOR CONDICIONES_FOR INCREMENTO_FOR) CUERPO_ESTRUCTURAS	El ciclo for "normal"	Permitir que usuario pueda definir un ciclo de tipo For	-
			65	ASIGNACION_FOR -> ASIGNACION_COMPLETA_FOR ASIGNACION ;	Los tipos de asignaciones que se pueden hacer en el apartado dedicado a ello del for	Permitir que un usuario pueda tener un tipo restringido de asignaciones a hacer en un ciclo for	-
			66	ASIGNACION_COMPLETA_FOR -> int nombre = CONTENIDO_NUMERO ; double nombre = CONTENIDO_NUMERO ; string nombre = CONTENIDO_STRING ; string nombre = CONTENIDO_VARIABLE ; char nombre = CONTENIDO_CHAR ; char nombre = CONTENIDO_VARIABLE ; boolean nombre = CONTENIDO_BOOLEAN ; boolean nombre = CONTENIDO_VARIABLE ; Object nombre = CONTENIDO_OBJECT ; Object nombre = CONTENIDO_VARIABLE ; Objeto nombre = CONTENIDO_OBJETO ; Objeto nombre = CONTENIDO_VARIABLE ;	El listado de las asignaciones que puede colocarse en un for	Permitir que el usauri tenga delimitados los tipos de asignaciones que puede realizaar en un for	-
			67	CONDICIONES_FOR -> CONTENIDO_BOOLEAN ; CONTENIDO_VARIABLE ; ;	Las dos formas en las que podría hacerse referencia a una condición en un for	Permitir que el usuario tenga delimitado el tipo de condición para definir los límites de operación de un ciclo for	-
			68	INCREMENTO_FOR -> OPERACION_UNARIA λ	El incremento que puede o no existir en la parte que tiene el for para realizar ello	Permitir que el usuario tenga definidas las formas que puede utilizar para hacer que el for avance	-
			69	INVOCACION -> CUERPO_INVOCACION_METODO ; this . CUERPO_INVOCACION_METODO ;	La estructura que sigue una invocación a un método o función en Java	Permitir que el usuario pueda hacer invocaciones a métodos o variables	-
			70	CUERPO_INVOCACION_VAR -> nombre . CUERPO_INVOOCACION_VAR nombre (ARGUMENTACION) . CUERPO_INVOCACION_VAR nombre	El listado que puede preceder a la variable invocada	Permitir que el usuario pueda invocar variables internas o externar únicas	-
			71	CUERPO_INVOCACION_METODO -> nombre . CUERPO_INVOOCACION_METODO nombre (ARGUMENTACION) . CUERPO_INVOCACION_MET nombre	El listado que puede preceder al método invocado	Permitir que el usuario pueda invocar métodos internos o externos	-
			72	ARGUMENTACION -> ARGUMENTOS λ	Permitir que la estructura de argumentos pueda o no aparecer entre los () de un método	Permitir que usaurio pueda apoyarse de los argumentos para pasar valroes a traves de los métodos	X
			73	ARGUMENTOS -> ARGUMENTOS , ARGUMENTO ARGUMENTO	La estructura que debe seguir un listado de argumentos de un método	Pemitir que el usuario peuda crear una lista de argumentos	-
			74	ARGUMENTO -> CONTENIDO_NUMERO CONTENIDO_STRING CONTENIDO_CHAR CONTENIDO_BOOLEAN CONTENIDO_OBJECT CONTENIDO_OBJETO	Los difernetes tipos de arguumentos que pueden ser enviados a un método	Permitir que el usuarip puea colocar los argumentos según los tipos válidsos en Java	X
			75	SENTENCIA_CONTROL -> SENTENCIA_IF SENTENCIA_IF else CUERPO_ESTRUCTURAS SENTENCIA_SWITCH	Los diferentes tipos de sentencias de control que existen en Java	Permitir que un usuario pueda utilizar cuanlquier sentencia de control existente en Java	X
			76	SENTENCIA_IF -> if (CONTENIDO_BOOLEAN) CUERPO_ESTRUCTURAS if (CONTENIDO_VARIABLE) CUERPO_ESTRUCTURAS	La sentencia de control IF, con las diferentes formas de representar una condición	Permitir que un usuario peuda definir correctamente una sentencias if	-
			77	CONTENIDO_BOOLEAN -> NEGACIONES (CONTENIDO_BOOLEAN) (CONTENIDO_BOOLEAN) CONTENIDO_BOOLEAN operador_logico CONTENIDO_BOOLEAN CONDICION	El contenido que conforma un tipo boolean junto con las agrupaciones y negaciones permitidas para definir a los mismos	Permitir que un usuario pueda definir contenido de tipo booleano en su proyecto	-
			78	CONDICION -> CONTENIDO_CONDICION operador_relacional OPCION_CONDICION CONTENIDO_BOOL operador_relacional OPCION_CONDICION CONTENIDO_BOOL	Las diiferntes formas de condición que pueden tener las condiciones compuestas	Permitir que un usuario pueda definir deiferntes formas de condiciones	-
			79	OPCION_CONDICION -> CONTENIDO_BOOL CONTENIDO_CONDICION	El tipo de contenido a comparar en la revision, cuando solo éste aparezca, es decir cuando corresponda a un valor booleano	Permitir que el usuario pueda presecindir de los tipos d e varlor booleanos para formas sus condiciones	-
			80	CONTENIDO_CONDICION -> CONTENIDO_NUMERO CONTENIDO_STRING CONTENIDO_CHAR CONTENIDO_OBJECT CONTENIDO_OBJETO	Todos los tipos de contenidos que se pueden comparar al momento de poner en acción una condición	Definir Iso direrentes tipo de contenido que pued haber en una condición simple	-

			81	CONTENIDO_BOOL -> (CONTENIDO_BOOL) NEGACIONES CONTENIDO_BOOL NEGACIONES CONTENIDO_VARIABLE booleano	El contenido booleano, como valor puro y variable con agrupación infinita permitida	Permitir que un usuario pueda definir un contenido booleano ya sea agrupado o no	-
			82	NEGACIONES -> NEGACIONES no no	Las negaciones	Permitir que el usuario pueda negar las condiciones y expresiones booleanas	-
			83	SENTENCIA_SWITCH -> switch (CONTENIDO_SWITCH) { CUERPO_SWITCH }	La estructura de control switch	Permitir que el usuario pueda definir estructuras condicionales de tipo switch	-
			84	CUERPO_SWITCH -> CASOS_NORMALES CASO_DEFAULT CASO_DEFAULT	El contenido propio del switch que puede tener al momento de implementarlo	Permitir que un usuario pueda agregar casos de estudio en el cuerpo de un switch	-
			85	CASOS_NORMALES -> CASOS_NORMALES CASO CASO	El listado de casos "normales" que un switch puede tener	Permitir que el usuario pueda conocer la estructura general que un caso normal debe seguir	-
			86	CASO -> case CONTENIDO_SWITCH : BLOQUE_GENERAL case CONTENIDO_SWITCH :	La estructura que un caso "normal" posee	Permitir que un usuario conozca la estructura que debe tener un caso listado en un switch	-
			87	CASO_DEFAULT -> default : BLOQUE_GENERAL default :	La estructura y contenido que un caso "default" posee	Permitir que un usuario pueda conocer la estructura que puede utilizar al momento de definir un caso Default	-
			88	CONTENIDO_SWITCH -> CONTENIDO_STRING CONTENIDO_NUMERO CONTENIDO_CHAR	El tipo de contenido que puede ser colocado dentro de los paréntesis de especificación de un switch	Permitir que el usuario conozca lo que puede incluir entre los paréntesis de un switch	-
			89	CUERPO_ESTRUCTURAS -> { BLOQUE_GENERAL } {}	La estructura general que el cuerpo de una estructura (sentencia de control, ciclo) puede poseer	Permitir que un usuario pueda definir un bloque	-
			90	BLOQUE_GENERAL -> BLOQUE_RETORNO BLOQUE RETORNO	La estructura que puede tener el bloque de una estructura de Java	Permitir que un usuario pueda agregar un bloque con sin o solamente un retorno en una estructura	-
			91	RETORNO -> RETORNO_COMPUUESTO parada ;	Los tipos de retorno que existen	Permitir que un usuario pueda definir un retorno de tipo compuesto dentro de sus estructuras	-
			92	RETORNO_COMPUUESTO -> RETORNO_NUMERO RETORNO_STRING RETORNO_CHAR RETORNO_BOOLEAN RETORNO_OBJECT RETORNO_OBJETO return INVOCACION return nombre . CUERPO_INVOCACION_VAR ; return this . CUERPO_INVOCACION_VAR ;	Los diferentes tipos de retorno con valor que se pueden realizar	Permitir que un usuario pueda definir los diferentes tipos de retornos que pueden existir dentro de las estructuras de tipo diferente a void	X
			93	RETORNO_VARIABLE -> return CONTENIDO_VARIABLE ;	La estructura que posee un tipo de retorno con contenido encerrado en una variable	Definir la estructura que un retorno de tipo variable puede poseer	-
			94	RETORNO_NUMERO -> return CONTENIDO_NUMERO ;	La forma que puede poseer un tipo de retorno numérico	Definir la estructura que un retorno de tipo número puede poseer	X
			95	RETORNO_STRING -> return CONTENIDO_STRING ;	La forma que puede poseer un tipo de retorno de Strings	Definir la estructura que un retorno de tipo String puede poseer	-
			96	RETORNO_CHAR -> return CONTENIDO_CHAR ;	La forma que puede poseer un tipo de retorno de char	Definir la estructura que un retorno de tipo char puede poseer	-
			97	RETORNO_BOOLEAN -> return CONTENIDO_BOOLEAN ;	La forma que puede poseer un tipo de retorno de boolean	Definir la estructura que un retorno de tipo boolean puede poseer	X
			98	RETORNO_OBJECT -> return CONTENIDO_OBJECT ;	La forma que puede poseer un tipo de retorno de Object	Definir la estructura que un retorno de tipo Object puede poseer	-
			99	RETORNO_OBJETO -> return CONTENIDO_OBJETO ;	La forma que puede poseer un tipo de retorno de Objetos personalizados	Definir la estructura que un retorno de tipo Objeto personalizado puede poseer	-
			100	PARADA -> return break	La forma que posee un return simple (sin valor a retornar) y una instrucción break	Definir la estructura que un break o parada puede poseer	-