

> AnalizadorLexico

◆ ANALIZADOR

INICIO (analizarLinea)

PARA caracterActual=0 **HACER** caracterActual++ **HASTA** caracterActual< tamaño(lineaDesglosada)

SI lineaDesglosada!= espacio en blanco

ENTONCES analizarAgrupaciones

FIN SI

SI hayQueAnalizarPrimitivos **Y** (caracterActual) <

tamaño(lineaDesglosada) & lineaDesglosada!= espacio en blanco

caracterActual = ultimaPosicionAnalizadaEnPrimitivos

FIN SI

SINO

caracterActual--

FIN

INICIO (analizarAgrupadores)

SI caracterActual != “)” **Y** tipoCaracter==c **Ó** listaEsperaCierre-> estaVacia

resultadosHallados = analizarAgrupaciones

tipoAgrupacion = tipoClasificacion

ENTONCES colorear

reestablecer tipoAgrupacion - >errónea

DEVOLVER ultimaLineaAnalizada + 1

FIN SI

DEVOLVER caracterActual

FIN

INICIO (analizarPrimitivos)

tipoCaracter = tipoCaracterDeterminado

ESCOGER según tipoCaracter

caso letra:

resultadosHallados = analizarOcurrenciaConLetras

parar

caso digito

resultadosHallados = analizarNumero

parar

caso otroCaracter Ó **caso** punto
resultadosHallados = analizarDemasCaracateres
parar

FIN ESCOGENCIA

colorear
reestablecer tipoAgrupacion -> erronea

DEVOLVER resultadosHallados

FIN

INICIO (analizarOcurrecenciaConLetras)

datosHallados = analisisPalabra

SI datosHallados !=erronea Y datosHallados !=caracter
tipo = busqueda de palabraReservadaDetablaSimbolos

SI tipo != erronea

SI tipoNuevo==Funcional

tipoAgrupacion = Funcional

FIN SI

SINO

tipoAgrupacion = tipoNuevo

SINO

excepcionMalformacionReservada

FIN SINO

datosHallados = tipo

FIN SI

resultado= datosHallados - 1

establecerFilaFin = numeroLinea

DEVOLVER resultado

FIN

INICIO(analizarPalabra)

posicionAnalisis = posicionInicialAnalisis + 1

parar = false

detallesPalabra = primitiva_palabra, concatenadoHastaElMomento +
caracterAdmision

MIENTRAS posicionAnalisis < tamaño de lineaAEstudiarY caracterActual!=
espacioEnBlanco Y parar == false

SI posicionAnalisis == (posicionInicialAnalisis + 1)

detallesPalabra = primitiva_palabra

FIN SI

SI determinarTipoCaracter != letra

detallesPalabra = excepcionPalabra

SI detallesPalabra==primitiva_palabra

DEVOLVER resultado de analizarIdentificador

FIN SI

SINO

parar = true

FIN SINO

FIN SI

SI parar == false

detallesPalabra+= caracterActual

posicionAnalisis++

FIN SI

FIN MIENTRAS

SI tamaño de detallesPalabra == 1

tipoAgrupacion = caracter

detallesPalabra = caracter

FIN SI

detallesPalabra= posicionAnalisis

DEVOLVER detallesPalabra

FIN

INICIO (analizarNumero)

hallazgos = hallazgos de analizarEntero

resultado = detalleHallazgos

resultado.establecerFilaFin= numeroLinea

DEVOLVERresultado

FIN

INICIO(analizarEntero)

posicionAnalisis = posicionInicialAnalisis + 1
detallesEntero = entero, caracterAdmision
parar = false

MIENTRAS posicionAnalisis < tamaño de lineaAEstudiar **Y** caracterActual!=
espacio en blanco **Y** parar == false

SI al determinarTipoCaracter != digito
detallesEntero = excepcionNumero
parar = true

SI detallesEntero == posible_decimal
DEVOLVER resultado de analizarDecimal
FIN SI

FIN SI

SI parar == false
detallesEntero += caracterActual
posicionAnalisis++

FIN SI

FIN MIENTRAS

detallesEntero = posicionAnalisis
tipoAgrupacion = entero

DEVOLVER detallesEntero

FIN

INICIO (analizarDecimal)

posicionAnalisis = posicionInicialAnalisis + 1
detallesDecimal = decimal, concatenadoHastaElMomento + caracterAdmision

MIENTRAS posicionAnalisis < tamaño lineaAEstudiar **Y** caracterActual !=
espacioEnBlanco

SI al determinarTipoCaracter != dígito
detallesDecimal = excepcionDecimal
PARAR MIENTRAS;

FIN SI

detallesDecimal += caracterActual
posicionAnalisis++

FIN MIENTRAS

SI detallesDecimal == decimal
 tipoAgrupacion = decimal
 detallesDecimal = posicionAnalisis
FIN SI

DEVOLVER detallesDecimal
FIN

INICIO (analizarSimbolosSimples)

datosSimboloSimple = posicionInicialAnalisis), caracterAEstudiar
nodoAuxiliar = primerSimbolo de TablaSimbolos

PARA simboloActual = 1 **HACER** simboloActual ++ **MIENTRAS** simboloActual <= tamaño listadoDeSimbolos

 simboloSimple = contenidoNodo

SI caracterAEstudiar == simboloSimple
 datosSimboloSimple = nombreNodo

SI datosSimboloSimple == símbolo igual **O** datosSimboloSimple == puntoComa
 tipoAgrupacion = finAsignacion

SINO
 tipoAgrupacion = símbolo

SIN FINO

SI datosSimboloSimple == parentesis_Cierre && último contenido de listaEsperaCierre empieza con "("
 eliminar último nodo de listaEsperaCierre

FIN SI

DEVOLVER datosSimboloSimple

FIN SI

 nodoAuxiliar = nodoSiguiente

FIN PARA

datosSimboloSimple = excepcionSimbolo

DEVOLVER datosSimboloSimple

FIN

INICIO (analizarSimbolosCompuestos)

conjuntoEstudio = caracterActual + caracterSiguiente

datosSimboloCompuesto = posicionInicialAnalisis + 1, conjuntoEstudio

nodoAuxiliar = primer nodo de listadoSimbolosCompuestos

PARA simboloActual = 1 **HACER** simboloActual ++ **MIENTRAS** simboloActual <= tamaño de listadoSimbolosCompuestos

simboloCompuesto = contenido de nodoAuxiliar

SI conjuntoEstudio == simboloCompuesto

datosSimboloCompuesto = nombre de nodoAuxiliar

tipoAgrupacion = simbolo

DEVOLVER datosSimboloCompuesto

FIN SI

nodoAuxiliar = nodoSiguiente

FIN PARA

datosSimboloCompuesto = excepcionSimbolo

DEVOLVER datosSimboloCompuesto

FIN

INICIO (analizarDemasCaracteres)

resultadosObtenidos

SI determinarTipoCaracter == punto

DEVOLVER resultado de analizarPunto

FIN SI

SI posicionInicialAnalisis < (tamaño lineaDeEstudio - 1) Y (caracterActual+ 1]) != espacioEnBlanco Y determinarTipoCaracter(caracterSiguiente) == otros

resultadosObtenidos = resultado de analizarSimbolosCompuestos

FIN SI

SINO

resultadosObtenidos = resultado de analizarSimbolosSimples

FIN SINO

resultado = datos de hallazgos

resultado establecerFilaFin = numeroLinea

DEVOLVER resultado

FIN

INICIO (analizarPunto)

SI al determinarTipoCaracter de caracterSienguiete == digito
 resultadosObtenidos = resultado de analizarDecimal
 resultado = datos de hallazgos del decimal

FIN SI

SINO

 resultado = datos erroneos

LLAMAR excepcionLexico de puntoDesubicado

FIN SINO

DEVOLVER resultado

FIN

❖ EXCEPCIONES

INICIO(excepcionPalabra)

SI determinarTipoCaracter de caracterActual == dígito Ó caracterActual == _
 DEVOLVER posible_identificador

FIN SI

DEVOLVER primitiva_palabra

FIN

INICIO(excepcionIdentificador)

 mensajeError = "hay caracteres incorrectos en en el conjunto de la columna " +
 columnaExcepcion

LLAMAR anadirError

return primitiva_palabra

FIN

INICIO(excepcionMalformacionReservada)

 mensajeError= el conjutno NO PERTENECE a palabras reservadas

LLAMAR anadirError

FIN

INICIO(excepcionNumero)

SI columnaExcepcion+1 < tamaño de lineaCompleta **Y** determinarTipoCaracter de carcaterExcepcion == punto

DEVOLVER posible_decimal
FIN SI

return primitiva_entero

FIN

INICIO (excepcionNecesitadosCierre)

LLAMAR añadirError

FIN

INICIO (excepcionDecimal)

SI determinarTipoCaracter de caracterExcepcion == punto
mensajeError = " el tipo decimal debe tener únicamente un PUNTO"
LLAMAR anadirError

DEVOLVER excepcionPorPuntoExtraEnDecimal
FIN SI

datosDecimal = primitiva_decimal
DEVOLVER datosDecimal

FIN

INICIO (excepcionPorPuntoExtraEnDecimal)

datosDecimalErrado = erronea

MIENTRAS posicion<tamaño de lineaDesglosada **Y** determinarTipoCaracter de caracterExcepcion == digito **O** determinarTipoCaracter de caracterExcepcion == punto

datosDecimalErrado +=demasCaracteresErrados

posicion++

FIN MIENTRAS

datosDecimalErrado = posicion
DEVOLVER datosDecimalErrado

FIN

INICIO (excepcionSimbolo)

mensajeError = La utilizacion deL "caracterErroneo"no está permitida

LLAMAR anadirError


```

    DEVOLVER erronea
FIN
INICIO (puntoDesubicado)
    mensajeError = Tienes un punto mal ubicado
    LLAMAR anadirError
FIN

INICIO (anadirError)
    mensajeError = Error en la linea: "numeroLinea+1" + mensaje
    LLAMAR añadir al final de listaErrores
FIN

```

◆ HERRAMIENTA

```

INICIO(determinarTipoCaracter)
    SI caracterInicial >= 65 Y caracterInicial <= 90) O )caracterInicial >= 97 Y
    caracterInicial <= 122
        DEVOLVER letra
    FIN SI

    SI caracterInicial >= 48 Y caracterInicial <= 57
        DEVOLVER digito
    FIN SI

    SI caracterInicial == 46
        DEVOLVER punto
    FIN SI

    SI caracterInicial == 34 O caracterInicial == 40 O caracterInicial == 47
        DEVOLVER requiereCierre
    FIN SI

    DEVOLVER otro
FIN

INICIO (agregarNumeroLineas)
    SI lineaPosicion> lineaMostrada
        DEVOLVER lineaMostrada++
    FIN SI

    DEVOLVER 0
FIN

```

INICIO (eliminarRangoLineas)

SI lineasLlenas < lineaMostrada

DEVOLVER (lineaMostrada-lineasLlenas)

FIN SI

DEVOLVER 0

FIN

INICIO (mostrarError)

nodoAuxiliar = LLAMAR darPrimerNodo de listaDeErrores

PARA errorActual=0 **HACER** errorActual++ **MIENTRAS** errorActual < tamaño de listaDeErrores

caja de texto añadir -> contenido de nodo

nodoAuxiliar = nodoSiguiente;

FIN PARA

FIN

INICIO (colorearConjuntoCaracteres)

ESCOGER (tipoAgrupacion)

caso entero:

colorear -> morado

FIN CASO

caso cadena:

colorear -> gris

FIN CASO

caso Funcional:

colorear -> verde

FIN CASO

caso decimal:

colorear -> celeste

FIN CASO

caso booleano:

colorear -> anaranjado

FIN CASO

caso caracter:

colorear -> marrón

FIN CASO

caso finAsignacion:

colorear -> rosado

FIN CASO

caso comentario:

colorear -> rojo

FIN CASO

caso simbolo:

colorear -> azul

FIN CASO

DE LO CONTRARIO:

colorear -> blanco

FIN CASO

FIN ESCOGENCIA

FIN