DRACO ORDENADO:

**Analizadores:**

De todos los módulos de un Compilador, ¿cuántos se dedican a la labor de análisis?

Hay tres módulos encargados del análisis: Analizador Léxico, Analizador Sintáctico y Analizador Semántico.

**Termina la frase:**

La primera fase de un compilador es el Análisis léxico.

**Transiciones:**

En el Autómata Finito correspondiente a un Analizador Léxico, si un determinado estado tiene una transición etiquetada con “d” (dígito) y otra etiquetada con "0" (cero), ¿cuál es la respuesta correcta?

- Habría que unir ambas transiciones en una única transición con "dígito", De esta manera, el Autómata dejaría de ser No Determinista

-El Autómata no es Determinista y no será válido

Explicación:De un mismo estado no pueden partir dos transiciones con el cero (el cero también es un dígito)

**Gramática operador**

-Si un Analizador Léxico tiene que reconocer el operador ++, ¿qué gramática sería la más adecuada?

S → + A

A → +

ó

A → + B

B → +

**No son acciones semánticas**

¿Cuáles de las siguientes son acciones semánticas que no son correctas en el diseño de un Analizador Léxico al reconocer la sentencia WHILE (a >= b AND b <= c) {a+= 33}

-1: Concatenar ('>'); Concatenar ('=')

-2: GenToken (ENTERO, -)

-3:InsertarTS (33)

-4: GenToken (IGUAL, -)

-5: GenToken (LLAVE, -)

-6:GenToken (ID, -)

-7:GenToken (ASIGNACIÓN, "a+= 33")

-8:buscarTS(‘+=’)

**Tokens varios 1**

Un lenguaje dispone de números reales, números enteros, cadenas y comentarios. ¿Cuáles de los siguientes podrían ser tokens de este lenguaje?

1. <FLOAT, valor>
2. <EOF, ->  
   Explicación:  
   Todo fichero tendrá su final de fichero, por lo que este token sería también correcto.
3. <ENTERO, valor>
4. <CADENA, posMemoria>  
   Explicación:  
   Corresponde a una codificación válida para una cadena, donde el atributo es un puntero a la posición de memoria donde se ha guardado la cadena.
5. <CADENA, lexema>  
   Explicación:  
   Corresponde a una codificación válida para una cadena.

**Tokens empresa -1**

A la hora de diseñar el conjunto de tokens de un Analizador Léxico para un fichero de personal de una empresa que contiene números de teléfono, nombres y apellidos, departamento y sueldo (cuyo formato es una serie de dígitos seguidos del signo de euro, por ejemplo, 3000€), ¿es correcto el siguiente diseño de los tokens?: <teléfono, lexema>, <palabra, lexema>, <Cod.Departamento,->, <Sueldo, lexema>

Falso → <Sueldo, valor>

**Fases de un compilador**

Ordena las siguientes fases de un proceso de compilación según el orden en que deben ocurrir.

1. **Análisis Léxico**  
   Explicación: Ésta es la primera fase del proceso de compilación, y va antes del Analizador Sintáctico
2. **Análisis Sintáctico**  
   Explicación: Ésta es la segunda fase del proceso de compilación, y va tras el Analizador Léxico y antes del Analizador Sintáctico
3. **Análisis Semántico**  
   Explicación: Ésta es la tercera fase del proceso de compilación, y va tras en Analizador Sintáctico y antes de iniciarse el proceso de Síntesis
4. **Generación de Código Intermedio**  
   Explicación: Ésta es la primera fase de la Síntesis en el proceso de compilación, y va tras el Analizador Semántico
5. **Optimizador de Código Intermedio.**

Explicación: Esta fase va tras el Generador de Código Intermedio

1. **Generación de Código Final**  
   Explicación: El código objeto es lo último que se genera en el proceso de compilación
2. **Optimizador de Código Objeto**  
   Explicación: Es lo último que se hace en un proceso de compilación y va tras haberse generado el código objeto.

**Tokens empresa - 2**

A la hora de diseñar el conjunto de tokens de un Analizador Léxico para un fichero de personal de una empresa que contiene números de teléfono, nombres y apellidos, departamento y sueldo (cuyo formato es una serie de dígitos seguidos del signo de euro, por ejemplo, 3000€), ¿es correcto el siguiente diseño de los tokens?: <teléfono, lexema>, <nombre, lexema>, <apellidos, lexema>, <Cod.Departamento,->, <Sueldo, valor>

Falso → <Palabra, lexema> en vez de <nombre, lexema>, <apellidos, lexema>

**Tokens empresa - 3**

A la hora de diseñar el conjunto de tokens de un Analizador Léxico para un fichero de personal de una empresa que contiene números de teléfono, nombres y apellidos, departamento y sueldo (cuyo formato es una serie de dígitos seguidos del signo de euro, por ejemplo, 3000€), ¿es correcto el siguiente diseño de los tokens?: <teléfono, valor>, <palabra, lexema>, <Cod.Departamento,->, <Sueldo, valor>

Falso → <teléfono, lexema> pues el teléfono no es un número; se debe enviar el lexema correspondiente a la concatenación de los dígitos

**Constante real**

¿Cual es el formato más correcto para el token constante real del Procesador de un Lenguaje de Programación?

<CTE\_REAL, valor>, <Número-Real, valor>  
Explicación:

Tanto el código como el atributo son adecuados

**Tokens empresa - 4**

A la hora de diseñar el conjunto de tokens de un Analizador Léxico para un fichero de personal de una empresa que contiene números de teléfono, nombres y apellidos, departamento y sueldo (cuyo formato es una serie de dígitos seguidos del signo de euro, por ejemplo, 3000€), ¿es correcto el siguiente diseño de los tokens?: <teléfono, lexema>, <palabra, lexema>, <Departamento, lexema>, <Sueldo, valor>

Falso → Pues normalmente los departamentos de una empresa son conocidos y sería más lógico identificar qué departamento es (Tabla de Departamentos) y enviar un token por cada departamento

**Token identificador**

¿Cuál es el formato más correcto para el token identificador del Procesador de un Lenguaje de Programación?

<Identificador, puntero a Tabla de Símbolos>  
<Identificador, posición en la Tabla de Símbolos>  
Explicación:

Es la mejor forma de representar a los identificadores

**Función del analizador léxico**

¿Cuáles de las siguientes actividades del Analizador Léxico?

1. Eliminar comentarios.
2. Eliminar espacios y tabuladores  
   Explicación:  
   Los espacios y tabuladores no aportan información al Procesador, por lo que deben eliminarse durante la lectura del fichero fuente.
3. Enviar los tokens al Analizador Sintáctico
4. Leer el fichero fuente carácter a carácter
5. Manejar el fichero fuente

Explicación:

Efectivamente, es el único módulo de un Procesador que gestiona el fichero de entrada

1. Relacionar los mensajes de error con el programa fuente  
   Explicación:  
   Como el Analizador Léxico es el único que maneja el fichero fuente, es él el que conoce la línea por la que va leyendo y por tanto se lo comunica al gestor de errores para que cuando dé un error pueda indicar la línea
2. Detectar los errores debidos a la escritura incorrecta de una palabra

**Termina la frase**

La segunda fase de un compilador es el Análisis sintáctico.

**Tokens empresa - 5**

A la hora de diseñar el conjunto de tokens de un Analizador Léxico para un fichero de personal de una empresa que contiene números de teléfono, nombres y apellidos, departamento y sueldo (cuyo formato es una serie de dígitos seguidos del signo de euro, por ejemplo, 3000€), ¿es correcto el siguiente diseño de los tokens?: <teléfono, índiceTS>, <palabra, lexema>, <Cod.Departamento,->, <Sueldo, valor>

Falso → Están mal pues los números de teléfono no deben ir en la Tabla de Símbolos

**Tokens varios - 2**

Un lenguaje dispone de identificadores, números enteros, operadores (+, -) y cada sentencia tiene que ir en una línea. ¿Cuáles de los siguientes podrían ser tokens de este lenguaje?

1. <ENTERO, valor>
2. <ID, ptrTS>
3. <MÁS, ->
4. <EOF, ->  
   Explicación:  
   Todo fichero tendrá su final de fichero, que es necesario mandarlo como token
5. <OP\_ARIT, "+">
6. <OP\_ARIT, menos>  
   Explicación:  
   Es una forma correcta de codificar una resta
7. <OP\_ARIT, 1>

Explicación:

Consideramos que el 1 representara al más o al menos

**Diseño de tokens**

A la hora de realizar el Analizador Léxico para un Lenguaje de Programación, indica cuál de los siguientes tokens es correcto.

<constante-entera, valor> ó <núm-entero, valor>  
Explicación:  
Las constantes o números enteros, independientemente de su notación, se deben identificar como tal ya que tienen una representación y operaciones diferentes de los reales

**Gramática de paréntesis**

Para gestionar que el número de paréntesis de apertura/cierre en un programa es correcto (están correctamente emparejados) en un lenguaje de programación genérico, ¿qué gramática para el Analizador Léxico debería diseñarse?

* Es imposible que la gramática de un Analizador Léxico pueda realizar esa comprobación.
* Es imposible representar el anidamiento correcto de los paréntesis con una gramática regular.

**Gramática**

Un lenguaje tiene: variables cuyo nombre se forma comenzando por una letra y pudiendo ir seguido de letras (l), dígitos (d) o subrayados (\_), pero no pudiendo terminar en subrayado; el operador suma (+) y el operador de autoincremento (++). Los elementos pueden ir separados por delimitadores. De los siguientes grupos de reglas (teniendo en cuenta que A es el axioma) ¿cuáles pertenecen a la gramática para el Analizador Léxico?

1. A → l B

B → l B | d B | \_ C | λ

C → l B | d B | \_ C

1. A → del A
2. A → + E

E → + | λ  
Explicación:  
Permite reconocer la suma y el autoincremento

**Orden tokens  
¿En qué orden se generarían los tokens para el siguiente fragmento de código fuente: while (a == 0) do print 5?**

1) <WHILE, ->

2) <PARENTabi, ->

3) <ID, ptrTS>

4) <IGUAL, ->

5) <ENT, 0>

6) <PARENTcerr, ->

7) <DO, ->

8) <PRINT, ->

9) <ENT, 5>

**REPASO LÉXICO (TOCHO)**

**PALABRAS RESERVADAS**

**¿Cuál de las siguientes afirmaciones es correcta en relación con las acciones semánticas del Analizador Léxico encargadas de generar el token palabra reservada**

Deben comprobar si el lexema leído coincide con una de las palabras reservadas del lenguaje para lo cual la pueden buscar en una Tabla de palabras reservadas.  
  
Deben comprobar si el lexema leído coincide con una de las palabras reservadas del lenguaje para lo cual la pueden buscar en la Tabla de Símbolos (donde se habrán metido previamente todas las palabras reservadas)  
Explicación:

Solo si el lexema corresponde con una de las palabras reservadas del lenguaje se puede generar dicho token, y guardarlas en la Tabla de Símbolos es una forma de hacerlo

Hay que comprobar si el lexema de la palabra leída coincide con una de las palabras reservadas del lenguaje.

**AUTÓMATA DEL LÉXICO**

**¿Cuáles de las siguientes actividades son función del Analizador Léxico?**

1: Manejar el fichero fuente

2: Leer el fichero fuente carácter a carácter

3: Enviar los tokens al Analizador Sintáctico

4. Eliminar comentarios

5. Eliminar espacios y tabuladores

6, Rellenar parte de la información de la Tabla de Símbolos

7. Detectar los errores debidos a la escritura incorrecta de una palabra

**FUNCIÓN DEL ANALIZADOR LÉXICO**

**¿Qué tipo de gramáticas tienen que utilizarse para construir un Analizador Léxico?**

Gramáticas Regulares

**GRAMÁTICA LÉXICA**

**Gramática del léxico**

¿Cuál de las siguientes gramáticas son válidas para un Analizador Léxico?

**S → a X | b X | λ**

**X → b X | a**

**Tokens empresa - 7**

A la hora de diseñar el conjunto de tokens de un Analizador Léxico para un fichero de personal de una empresa que contiene números de teléfono, nombres y apellidos, departamento y sueldo (cuyo formato es una serie de dígitos seguidos del signo de euro, por ejemplo, 3000€), ¿es correcto el siguiente diseño de los tokens?: <teléfono, lexema>, <palabra, lexema>, <Cod.Departamento,->, <Sueldo, índiceTS>

**Falso**

**-----------------------------------------------------------------------------------------**

**Acciones semánticas**

¿Cuáles de las siguientes podrían ser acciones semánticas que ejecutaría un Analizador Léxico al reconocer la sentencia IF (88 != 3.321) limite:= "hola"?

1. GenToken (PARÉNTESISdcho, -)
2. GenToken (CADENA, "hola")
3. GenToken (DISTINTO, -)
4. GenToken (ID, posTS)
5. Concatena
6. BuscarTS (palabra)
7. InsertarTS (palabra)
8. valor:= valor \* 10 + d  
   Es necesario ir calculando el valor del número conforme se está leyendo para enviarlo en el token cuando se termine el reconocimiento del número
9. Para las transiciones no previstas en el autómata, se ejecutaría una acción de ERROR
10. GenToken (IF, -)
11. Leer

**Gramática de operadores y números**

Un lenguaje permite realizar sumas, restas, productos y divisiones de números enteros. ¿Cuáles de las reglas siguientes tendrá la gramática para un Analizador Léxico? (del es el blanco y tabulador; S es el axioma.)

1)S → + | - | \* | / | d A | del S

A → d A | λ

2)S → d A | \* | / | + | - | del S

A → d A | λ

3)S → d A | \* B | / C | + D | - E | del S

A → d A | del

B → del

C → del

D → del

E → del

**Gramática del Léxico**

S → a X | b X

X → b X | a

**Acciones semánticas**

1: Leer

2. GenToken (ENTERO, 88)

**Autómata del Léxico**

¿Cómo tiene que ser un Autómata de un Analizador Léxico?

Autómata Finito Determinista ó AFD permiten reconocer lenguajes regulares

**Comentarios**

¿Cuál de las siguientes afirmaciones es correcta en relación con el tratamiento de los comentarios en un lenguaje de programación? → Nunca hay que generar un token comentario

El Analizador Léxico no generará un token para los comentarios  
El Analizador Léxico se saltará todo el comentario sin generar un token para el comentario.

El Analizador Léxico se saltará todo el comentario y continuará su análisis hasta que encuentre el siguiente token válido.

**Gramática bis**

Un lenguaje tiene: variables cuyo nombre se forma comenzando por una letra y pudiendo ir seguido de letras (l), dígitos (d) o subrayados (\_), pero no pudiendo llevar dos subrayados seguidos; números enteros en binario (b); y el operador de igualdad (==). Los elementos tienen que ir separados por delimitadores. De los siguientes grupos de reglas (teniendo en cuenta que A es el axioma) ¿cuáles pertenecerían a la gramática para el Analizador Léxico?

1:

A → del A

2:

A → = E

E → = F

F → del

3:

A → b H

H → b H | del

4.

A → l B

B → l B | d B | \_ C | del

C → l B | d B | del

**Acción semántica LEER**

¿Cuál de las siguientes afirmaciones es cierta en relación con la acción semántica LEER (car:= leer ())?

1. Devuelve el siguiente carácter del fichero fuente
2. No se ejecuta si el carácter que teníamos ya leído (y que se ha usado para transitar) no forma parte del token que se está reconociendo.  
   Explicación:  
   Si el carácter ya leído no forma parte del token no hay que avanzar al carácter siguiente (porque entonces lo habríamos perdido). Se trata de un carácter que está indicando que el token ha finalizado pero que no se ha equiparado con el patrón del token que se está reconociendo (no se ha consumido), y por tanto es el inicio del token siguiente (o algún carácter irrelevante como un blanco, inicio de comentario...). Será el carácter que reciba el Analizador Léxico como entrada la siguiente vez que se le llame
3. Permite al analizador léxico disponer del siguiente carácter, una vez que el carácter previo ya se ha visto que forma parte del token que se está analizando (ya se ha consumido).
4. Antes de comenzar el análisis léxico de un fichero hay que inicializar el módulo del Analizador Léxico leyendo el primer carácter del fichero

**Tokens empresa - 6**

A la hora de diseñar el conjunto de tokens de un Analizador Léxico para un fichero de personal de una empresa que contiene números de teléfono, nombres y apellidos, departamento y sueldo (cuyo formato es una serie de dígitos seguidos del signo de euro, por ejemplo, 3000€), ¿es correcto el siguiente diseño de los tokens?: <teléfono, lexema>, <palabra, posiciónTS>, <Cod.Departamento,->, <Sueldo, valor>

**Falso** →

**PALABRAS RESERVADAS**

**¿Cuál de las siguientes afirmaciones es correcta en relación con las acciones semánticas del Analizador Léxico encargadas de generar el token palabra reservada**

* Deben comprobar si el lexema leído coincide con una de las palabras reservadas del lenguaje para lo cual la pueden buscar en la Tabla de Símbolos (donde se habrán metido previamente todas las palabras reservadas).
* Hay que comprobar si el lexema de la palabra leída coincide con una de las palabras reservadas del lenguaje.

**Concatenar**

**¿Cuál de las siguientes afirmaciones es cierta en relación con la acción semántica CONCATENAR?**

Se ejecuta solo si el carácter forma parte del lexema del token que se está reconociendo

Va almacenando en una variable el lexema que el analizador está leyendo

Permite al analizador léxico disponer del lexema del token reconocido (secuencia de caracteres que lo forman)

No puede ser la última acción a ejecutar

**Lenguajes sin declaración -1**

Declarada - falso

**Gramática y acciones semánticas de fechas**