



---

### Normas del examen

- Se dispone de **3 horas** para su realización.
  - La entrega debe consistir en una serie de hojas de examen, debidamente **numeradas**, con las respuestas del alumno. Todas las hojas deben llevar los **apellidos**, el **nombre** y el número de **D.N.I.** del alumno.
  - El examen consta de dos partes que deberán entregarse en **hojas independientes** para facilitar su corrección. La parte I corresponde a las preguntas 1, 2 y 3 (sobre el primer cuatrimestre) y la parte II a las respuestas de las preguntas 4, 5 y 6 (sobre el segundo cuatrimestre).
  - Para superarlo con éxito es necesario obtener una calificación **igual o superior a 5 puntos**.
- 

### PARTE I

1. **[1 punto]** Formaliza mediante un autómata finito determinista optimizado la siguiente descripción de una categoría léxica que se necesita para reconocer ciertas direcciones de correo electrónico.

Una *dirección de correo electrónico* típica siempre comienza con un *nombre*, compuesto por *letras* (mayúsculas o minúsculas), *dígitos decimales* y *puntos*, aunque estos últimos no pueden aparecer de manera consecutiva y ni al principio ni al final del nombre. El nombre debe tener un mínimo de tres caracteres de longitud y debe empezar siempre por una letra. A continuación es obligatorio que aparezca una arroba @ y el *servidor*, estando este último compuesto solamente por letras y dígitos decimales, y teniendo una longitud mínima de dos caracteres. Finalmente el punto . y el *dominio* deben aparecer inmediatamente después, estando este último compuesto únicamente por letras y teniendo una longitud comprendida entre los dos y los cuatro caracteres.

2. **[1 punto]** Acondiciona la siguiente gramática de atributos, eliminando la recursión a izquierdas y factorizando las producciones que puedan resultar problemáticas para la implementación del correspondiente analizador descendente predictivo recursivo.

Valve ::= Beep * Valve	{Valve <sub>0</sub> .nt = Beep.t * Valve <sub>1</sub> .nt}
Valve ::= Beep Up Down	{Valve.nt = func(Beep.t, Up.t, Down.t)}
Torque ::= Torque    Beep	{Torque <sub>0</sub> .nt = Torque <sub>1</sub> .nt   Beep.t}
Torque ::= High	{Torque.nt = High.t * 2}
Torque ::= Low	{Torque.nt = Low.t / 2}

3. **[1 punto]** Traduce el siguiente programa a pseudocódigo P para una máquina a pila con contador de instrucciones, según la arquitectura y los esquemas de traducción estudiados en el primer parcial.

```
edad: natural;
ciclos: natural;
show: char
&
if edad > 18 then
  while ciclos > 0 do
    begin
      out(show);
      ciclos := ciclos - 1
    end
```

## PARTE II

4. [3 puntos] Considera el siguiente programa:

```
tipo tpuntint = ^int;
tipo tfila = array [10] of tpuntint;
tipo tpuntfila = ^tfila;
tipo tdatos = array [10] of tpuntfila;
tipo tmatriz = reg f: int; c: int; datos: tdatos; freg;

matriz: tmatriz;
nfilas: int;
ncols: int;

procedure creaMatriz (var m: matriz, filas: int, cols: int) {
    i: int;
    j: int;
    &
    m.f := filas;
    m.c := cols;
    for i:=0 to filas do {
        new(m.datos[i]);
        for j:=0 to cols do
            new (m.datos[i]^[j]);
    }
}

procedure rellena (var m: matriz, i:int, j:int, v: int){
    procedure rellenaCelda(var pelda: tpuntint, v:int) {
        pelda^:=v;
    }
    puntcelda : tpuntint;
    & (*2*)
    puntcelda := m.datos[i]^[j]; (*3*)
    rellenaCelda( puntcelda, v); (*4*)
}
(*1*)
&
nfilas := 2;
ncols := 5;
creaMatriz(matriz,nfilas,ncols);
rellena(matriz,1,3,8);
```

- [0.5 puntos] Escribe el contenido de la tabla de símbolos en el punto (\*1\*). Considera que los tipos básicos (enteros y punteros) ocupan una posición de memoria.
- [1 punto] Haz un esquema detallado de la memoria de la máquina P en el punto (\*2\*). Incluye todo el detalle posible sobre los marcos de activación, memoria dinámica y contenido de las celdas. Considera que la memoria tiene únicamente 1000 posiciones ([0,999]).
- [0.75 puntos] Escribe el código P necesario para realizar la asignación indicada por (\*3\*).
- [0.75 puntos] Escribe el código P necesario para realizar la llamada al procedimiento y el paso de parámetros en la invocación al método rellenaCelda() (\*4\*).

5. [2 puntos] Considera la siguiente gramática incontextual:

$A \rightarrow BCD$

$B \rightarrow aCb$

$B \rightarrow \lambda$

$C \rightarrow cAd$

$C \rightarrow eBf$

$C \rightarrow gA$

$C \rightarrow \lambda$

$D \rightarrow b$

- Encuentra los conjuntos primeros y siguientes de sus no terminales
- Calcula el conjunto de directores. ¿Es ésta una gramática LL(1)? Justifica tu respuesta.
- Si fuera LL(1), muestra cómo se realizaría el análisis de la cadena “abefb” por medio del analizador descendente predictivo guiado por tabla.

6. [2 puntos] Considera la siguiente gramática incontextual:

$S \rightarrow AaB$

$S \rightarrow B$

$A \rightarrow b$

$A \rightarrow cB$

$B \rightarrow A$

- Encuentra el autómata LR(1) de esta gramática.
- ¿Es una gramática SLR(1)? Justifica tu respuesta.
- Muestra cómo se realizaría el análisis de la cadena “cbab” por medio del analizador LR dirigido por tabla.