

7. **3.4 puntos** Dada una empresa que dispone de una serie de teléfonos móviles asignados a sus trabajadores, se pretende realizar un control tanto del consumo mensual como del número de llamadas que se hacen por cada línea (con independencia de la duración de la llamada). Con ese propósito, la compañía de teléfono facilita cada mes a la empresa un documento en la que se detallan todas y cada una de las llamadas realizadas de cada móvil. Un ejemplo del detalle para un teléfono móvil es el siguiente:

```
Total Factura, 185.10, 639121190, Plan tarde

2007-04-04,13:59,651345045,Promoción,Alphatel,4:53,Normal,,0.1200
2007-04-04,17:04,678686082,Nacional,Móvil,2:43,Super R.,,0.4760
2007-04-04,18:42,686456441,Promoción,Alphatel,1:00,Normal,,0.1200
2007-04-05,18:05,678626034,Nacional,Móvil,4:29,Super R.,,0.6880
2007-04-05,19:44,622441670,Promoción,Alphatel,1:17,Normal,,0.1200
2007-04-05,21:03,633686089,Nacional,Móvil,0:37,Super R.,,0.2240
2007-04-06,11:40,666377047,Mensaje Corto,Alphatel,0:00,Normal,,0.1500
2007-04-07,11:43,699758805,Promoción,Alphatel,1:00,Normal,,0.1200
2007-04-30,21:21,958280654,Provincial,Granada,1:19,Super R.,,0.2290
2007-04-02,14:59,617222225,Núm. Alphatel,Alphatel,2:14,Normal,,0.7977
.....
```

Como se puede apreciar, aparece una cabecera y luego el detalle de las llamadas realizadas. Los campos de la cabecera la componen un texto indicativo, el consumo total y posteriormente aparece el número de teléfono móvil y el plan asignado

El detalle de las llamadas aparece justo debajo con los siguientes campos separados por comas:

- **Fecha:** formato numérico AAAA-MM-DD.
- **Hora:** formato numérico HH:MM.
- **Nº Receptor:** 9 cifras del número de teléfono al que se llama.
- **Tipo:** uno de los siguientes: Provincial, Nacional, Num. Alphatel, Mensaje corto y Promocion.
- **Destino:** uno de los siguientes: Alphatel, Móvil y Provincia (nombre de la provincia destinataria).
- **Tarifa:** uno de los siguientes: Normal y Super R.

Se desean realizar las siguientes comprobaciones:

- Determinar si el consumo total/mes de un teléfono supera la cantidad de **150 euros**.
- En base al horario en el que realiza las llamadas, determinar el plan de ahorro más apropiado en cada móvil. Si existe un balance favorable de llamadas en horario de mañana, de tarde o no es significativa la diferencia. Según lo anterior, existen tres planes de ahorro disponibles en la empresa de móviles:
 - **Plan mañana:** precio reducido en llamadas realizadas en la franja que va desde las 08:00h hasta las 13:00h.
 - **Plan tarde:** precio reducido en llamadas realizadas en la franja que va desde las 17:00h hasta las 22:00h.
 - **Plan constante:** precio único en todas las llamadas con independencia del horario.

En base a la descripción anterior y los requisitos expuestos:

- (a) **1.4 puntos** Definir el lenguaje del documento anterior mediante una gramática con las producciones en forma gramatical de tal forma que si existe algún error en la lectura de los campos que no son determinantes, es decir, en todos excepto en *Total Factura* correspondiente a la cabecera y en *Hora* del detalle de las llamadas, se pueda recuperar y continuar analizando.

Solución:

En primer lugar definiremos el lenguaje sin las producciones de error. De esta forma, vamos a considerar el lenguaje sin ningún tipo de abstracción, si bien, es posible definirlo en base a la abstracción solicitada en el siguiente apartado.

El lenguaje quedaría así:

```

moviles → moviles movil
        | movil
movil → cabecera llamadas
cabecera → totalfactura "," numero "," plan
totalfactura → "Total Factura," precio
numero → numero num
        | num
num → 0 | 1 | ... | 9
plan → "Plan mañana" | "Plan tarde" | "Plan constante"
precio → numero "." numero
llamadas → llamadas llamada
        | ε
llamada → fecha "," hora "," numero "," tipo "," destino "," hora "," horario "," precio
fecha → numero "-" numero "-" numero
hora → numero ":" numero
tipo → "Promocion" | "Nacional" | "Mensaje Corto" | "Provincial" | "Num. Alphatel"
destino → "Alphatel" | "Móvil" | "Álava" | ... | "Zaragoza"
horario → "Normal" | "Super R."

```

A continuación, añadimos las producciones de error en aquellos lugares en los que es necesario recuperarse. Según el problema, en cualquiera de los campos anteriormente especificados excepto en `precio` de la producción que define al símbolo no terminal `totalfactura` y en la primera aparición de `hora` de la producción `llamada`.

Con esas premisas, podemos permitir un error sintáctico en el campo de texto del total de la factura pero no en el precio final, es decir, quedaría la siguiente producción:

totalfactura	→	texto precio
texto	→	"Total Factura,"
		error

También podemos permitir errores sintácticos en los detalles de una llamada a excepción de la hora en la que se efectúa la llamada. De esta forma, la descripción gramatical quedaría así:

llamada	→	fecha hora restollamada
fecha	→	numero "-" numero "-" numero
		error
restollamada	→	"," numero "," tipo "," destino "," hora "," horario "," precio
		error

- (b) 1.0 puntos Definir los tokens con el máximo nivel de abstracción para contemplar las comprobaciones anteriormente mencionadas y obtener la gramática abstracta correspondiente.

Solución:

El problema nos dice que la auditoría en las llamadas presenta dos requisitos fundamentales. El primero es observar el precio total gastado en el mes y en segunda instancia la determinación del plan de llamadas que sea más apropiado. Para lo primero debemos observar el campo `PRECIO` y para lo segundo debemos prestar atención, dentro del detalle de la llamada, a la hora a la cual se realiza la misma. El resto de la información no es necesaria para las consideraciones semánticas, en cuyo caso, todo lo que se pueda unir formando gramática regular se podrá realizar sin ningún tipo de problema.

En estas circunstancias, la tabla de tokens quedaría así:

Nº	Token	Patrón
1	TOTALFACT	"Total Factura,"
2	PLAN	"Plan mañana" "Plan tarde" "Plan constante"
		atrib=0, atrib=1, atrib=2
3	PRECIO	{dig}+"."{dig}+
4	HORA	{dig}+":"{dig}+
5	FECHA	{dig}+"-"{dig}+"-"{dig}+
6	RESTOLLAM	{restollam}
7	NUMMOVIL	{dig}+

Para una mayor comprensión, se muestra la descripción de esta tabla de tokens según la sintaxis de LEX/FLEX:

```
%{
char *numMovil; // variable que almacena el número del móvil.
int consumoMes; // variable que almacena el consumo.

extern yylval;
%}

%option noyywrap

dig      [0-9]+
nhora    [0-9]{2,2}
nyear    [0-9]{4,4}
nummovil [0-9]{9,9}
precio   {dig} "." {dig}
hora      {nhora} ":" {nhora}
fecha     {nyear} "-" {nhora} "-" {nhora}
tipo      ("Nacional"|"Promocion"|"Mensaje Corto"|"Provincial"|"Num. Alphanet")
destino   ("Movil"|"Alphanet"|"Granada")
horario   ("Normal"|"Super R.")
restollam {nummovil} "," {tipo} "," {destino} "," {hora} "," {horario} "," {precio}

%%
[ \t\n]+ ;
"Total Factura," { return TOTALFACT ; }
"Plan mañana" { yylval= 0; return PLAN ; }
"Plan tarde" { yylval= 1; return PLAN ; }
"Plan constante" { yylval= 2; return PLAN ; }
{precio} { consumoMes= atoi(yytext); return PRECIO ; }
{hora} { return HORA ; }
{fecha} { return FECHA ; }
{restollam} { return RESTOLLAM ; }
{nummovil} { numMovil= strdup (yytext); return NUMMOVIL ; }
. { printf("[Linea %d] Error en %s\n", yylineno, yytext); }
%%
```

La gramática abstracta resultante, según la sintaxis de YACC y sin considerar las producciones de error quedaría así:

```
%{
#include <stdio.h>
%}

%token TOTALFACT PLAN PRECIO HORA FECHA RESTOLLAM NUMMOVIL
%%

moviles      : moviles movil
              | movil ;
movil        : cabecera llamadas ;
cabecera     : totalfactura PLAN ;
totalfactura : texto PRECIO ;
texto        : TOTALFACT NUMMOVIL
llamadas     : llamadas llamada
              | ;
llamada      : FECHA HORA RESTOLLAM ;

%
#include "lex.yy.c"

int main (void)
{
    return yyparse();
}
```

- (c) 1.0 puntos Definir la gramática con atributos en sintaxis de YACC a partir de la gramática anterior para que contemple las comprobaciones que se exigen en el problema.

Solución:

Aunque se puede plantear una solución utilizando una tabla de almacenamiento temporal (TS), también es posible una solución que no considere dicha estructura dada la forma en la que se representan las llamadas para cada teléfono móvil. En este caso, la solución sería más simple dado que con el uso de variables que representen el total de llamadas en cada franja horaria se podría llevar dicha contabilidad. Dichas variables se tendrían que reiniciar al aparecer el detalle de facturación de

un nuevo número de teléfono.

Vamos a suponer que existe una función que analice la hora en la que se ha realizado la llamada y nos determine si pertenece al plan de mañana o al plan de tarde:

- `int PlanDeLlamada (char *hora):` Devuelve 0 si se ha realizado en horario del plan de mañana y 1 si fue en horario de plan de tarde.

Con toda esta información, tomando la gramática anterior sin las producciones de error, la gramática con atributos quedaría así:

```
%{
#include <stdio.h>

#define MAXCONSUMO 150 // Máximo consumo que se tolera.
#define PLANMAN      0  // Plan de mañana.
#define PLANTAR      1  // Plan de tarde.

int llamDia ;           // Número de llamadas en horario de mañana.
int llamTarde ;         // Número de llamadas en horario de tarde.
int planInicial ;       // Plan de llamadas que tiene asignado.
%}
%token TOTALFACT PLAN PRECIO HORA FECHA RESTOLLAM NUMMOVIL
%%

moviles          : moviles movil
                  | movil ;
movil             : cabecera { llamDia= 0; llamTarde= 0; }
                  llamadas
                  {
                      // INFORME DEL NÚMERO DE LLAMADAS REALIZADAS
                      if (planInicial== PLANMAN)
                      {
                          if (llamDia < llamTarde)
                              printf ("\nNº Movil: %s, se aconseja cambiar
                                  al Plan de Tarde\n", numMovil);
                      }
                      else // plan es de tarde
                      {
                          if (llamDia > llamTarde)
                              printf ("\nNº Movil: %s, se aconseja cambiar
                                  al Plan de Mañana\n", numMovil);
                      }

                      if (llamDia == llamTarde)
                          printf ("\nNº Movil: %s, se aconseja cambiar
                                  al Plan Constante\n", numMovil);
                  }

cabecera          : totalfactura PLAN          { planInicial= $2; }
totalfactura      : texto PRECIO
texto             : TOTALFACT NUMMOVIL
llamadas          : llamadas llamada
                  | ;
llamada           : FECHA HORA RESTOLLAM      { int plan= PlanDeLlamada ($2);
                                              if (plan==PLANMAN) // Horario de mañana.
                                                  llamDia++;
                                              else if (plan=PLANTAR)
                                                  llamTarde++;
                                              }

%%
```