



2. (3,5 puntos) Se desea automatizar el control de una red de aparcamientos gestionados por la misma empresa municipal, de acuerdo a los siguientes requisitos:

- Los usuarios del aparcamiento dispondrán de una tarjeta magnética donde figura registrado su código de identificación. Con este código, se puede identificar el resto de información personal y de uso del usuario.
- A su llegada a uno de los aparcamientos, el usuario introducirá la tarjeta en el lector correspondiente, lo que hace que se eleve la barrera situada en la entrada. Esta barrera permanece levantada un cierto tiempo, descendiendo automáticamente una vez ha pasado el vehículo o se ha cumplido un tiempo.
- Para salir del aparcamiento se procede de igual forma con la barrera situada a la salida.
- Tanto las entradas como las salidas deben quedar registradas con objeto de realizar periódicamente una facturación a los usuarios, según el tiempo de aparcamiento consumido. Estas facturas se emitirán a petición del operador bien semanal, mensual o trimestralmente.
- En caso de que algún sistema de entrada o salida esté averiado, estas tareas las realizará manualmente un empleado de la empresa, que deberá contabilizar en el sistema dicha actividad para mantener la facturación.

El sistema debe tener en cuenta la ocupación de cada aparcamiento, controlando un semáforo situado a la entrada. Si hay plazas libres, el semáforo debe estar verde, pasando a rojo si el aparcamiento se llena. Cuando el aparcamiento esté lleno no debe permitirse la entrada de nuevos vehículos.

El Departamento de Gestión Económica es el encargado de regular la facturación y cobro de los servicios del establecimiento. En caso de que un cliente no pague su factura tras un tiempo límite tras su recepción, se le suspenderá automáticamente su condición de usuario, pasando el expediente al Departamento de Morosos, que iniciará las correspondientes acciones para reclamar la cantidad adeudada.

Se desea:

- a) Desarrollar parcialmente la especificación de requisitos funcionales. Para ello, haga uso de *diagramas de casos de uso en UML 2*.
Se anulará todo aquel diagrama que no se presente de forma clara y pulcra.
- b) Analice qué tipo de componentes arquitectónicos serían los más adecuados para este sistema (o partes del sistema), según las recomendaciones de Bosch. Estudie primero cada requisito no funcional por separado y, posteriormente, exponga sus conclusiones sobre el conjunto.

3. (2,5 puntos) Utilizando como base un mismo programa pero en versiones iterativa y recursiva, se pide realizar una comparativa de ambas propuestas. Para ello, realice el conteo de ambas porciones de código aplicando estrictamente los criterios de Christensen *et al.* Indique para cada código:

- a) Para cada código, responda según los resultados obtenidos aplicando Halstead:
- ¿Cuántos bits son necesarios para representar un término de su vocabulario?
 - ¿Cuántos se requieren para el volumen del programa completo?
 - ¿Qué dificultad se obtiene para cada versión?
 - ¿Qué esfuerzo ha requerido este desarrollo?
 - ¿Qué información funcional útil contiene cada código?
- b) En base a las métricas de Halstead obtenidas en el apartado anterior, realice una comparativa razonada de ambos códigos y analice los motivos por los que se producen estos resultados.
- Se anulará cualquier respuesta que no sea convenientemente razonada.

```
static int fibonacci (int n)
{
    if ((n == 0) || (n == 1))
        return 1;
    else
        return fibonacci(n-1) + fibonacci(n-2);
}
```

A. Código Fibonacci recursivo

```
static int fibonacci (int n)
{
    int actual, ant1, ant2;
    ant1 = ant2 = 1;
    if ((n == 0) || (n == 1)) {
        actual = 1;
    } else
        for (i=2; i<=n; i++) {
            actual = ant1 + ant2;
            ant2 = ant1;
            ant1 = actual;
        }
    return actual;
}
```

B. Código Fibonacci iterativo