

## Fundamentos de Programación

PEC9 - 20191

Fecha Límite de entrega: 16/12/2019

Estudiante

**Apellidos:** DELGADO FLORES

**Nombre:** PABLO JOSÉ

### Objetivos

- Comprender que son los tipos abstractos de datos (TAD) y saber definirlos correctamente.
- Saber identificar el TAD más adecuado para cada problema y justificar su elección.

### Formato y fecha de entrega

Hay que entregar la PEC antes del día **16 de diciembre de 2019** a las 23:59.

Para la entrega se deberá entregar un fichero que contenga:

- Este documento con la respuesta de los diferentes ejercicios planteados.

Hay que hacer la entrega en el apartado de entregas de EC del aula de teoría.

## Enunciado

Siguiendo con la ayuda que proporcionamos a UOCBookings, la compañía nos ha pedido, ahora, nuestra colaboración para realizar un análisis respecto a los tipos abstractos de datos (TADs) más adecuadas para gestionar aspectos de la entrada y la salida de los clientes en los hoteles.

Por ello, en esta PEC, nos centraremos en los aspectos de análisis de los problemas planteados, dejando de lado los aspectos de codificación e implementación. Por este motivo, hay que responder a los ejercicios de este PEC en forma de texto y sólo en lenguaje algorítmico cuando lo requiera el enunciado. Se trata, sobre todo, de justificar las respuestas dadas.

### Ejercicio 1: (20%)

Apartado 1.a: (50%): Los hoteles deben llevar a cabo un control de los clientes que llegan en una fecha en concreto. Cuando se hace una reserva, se acostumbra a solicitar la hora aproximada de llegada. La relación de clientes deberá seguir un orden según la hora prevista de llegada. También puede pasar que a última hora se produzcan cancelaciones. Con estas cancelaciones se debe actualizar el registro de entrada del día (ya que los clientes finalmente no irán al hotel y deben ser eliminados del registro). Razonad que TAD será la más apropiada para llevar a cabo el control de clientes esperados en una fecha.

## SOLUCIÓN

**En este caso, utilizaría un TAD lista ya que puede suceder que en determinado momento un cliente cancele su reserva, por lo que necesito poder actualizar dichas reservas independientemente del lugar en el que el cliente esté posicionado.**

**Hay que recordar que en el caso de usar un TAD pila o un TAD cola, no se podrá actualizar la lista de reservas en el caso de que no se trate del primer elemento.**

Apartado 1.b: (50%): Para las salidas previstas en una fecha en concreto, el hotel también necesita tener la relación de clientes que saldrán. Hay que tener en cuenta que esta relación también puede sufrir alteraciones de última hora, ya que algunos clientes pueden solicitar dejar la habitación el mismo día, pero más tarde (*late checkout*) o, incluso alargar su estancia algún día más si el hotel tiene disponibilidad, cosa que implicaría su eliminación del registro de salidas del día. Razonad que TAD es el más apropiado para este control.

## SOLUCIÓN

Nos encontramos en el mismo caso que el anterior, dado que los clientes tienen la libertad de alterar su salida seleccionando un *late checkout* o alargar su estancia. Es por esto que el TAD recomendado vuelve a ser una lista.

### Ejercicio 2: (20%)

Asociada a una reserva tenemos también información respecto a la política de cancelación de la misma. Esta información es única por reserva y dependerá de las condiciones que haya contratado el cliente. La política de cancelación debe contemplar qué modificaciones admite la reserva. Por ejemplo, podemos ir desde la situación más restrictiva, en la que la reserva no se puede modificar en absoluto y su cancelación comporta la pérdida del importe pagado, hasta la más flexible en la que se puede cancelar totalmente hasta el día antes sin coste añadido. Entre estas dos situaciones extremas, podemos encontrar reservas que permiten cambiar fechas, pero no cancelarla o reservas que permiten la cancelación sin coste hasta una cierta fecha, y pérdida de importe pasada dicha fecha.

Vuestra tarea consiste en definir una estructura de datos, no necesariamente un TAD, que permita registrar la información sobre las condiciones de cancelación asociadas a una reserva en concreto y la relacione con la reserva. Si llamamos *tBookingChanges* a dicha estructura de datos, indica qué información crees que debería incluir. Debéis justificar la respuesta.

NOTAS: Para acotar el ejercicio suponed que hay cuatro tipos de cancelaciones posibles:

- La reserva no se puede modificar.
- La reserva se puede anular sin coste hasta una fecha en concreto, a partir de la cual se abonará el importe total.
- La reserva no se puede anular, pero permite cambiar las fechas.
- La reserva se puede anular con una multa que consiste en un porcentaje sobre el coste total.

Podéis suponer que cada reserva tiene un identificador de tipo entero en su estructura de datos.

## SOLUCIÓN

En esta situación, la estructura de datos que utilizaría sería un *enum* y una *tupla*, dado que con el *enum* podría definir las cuatro posibles cancelaciones:

1. La reserva no se puede modificar
2. La reserva se puede anular sin coste hasta una fecha en concreto, a partir de la cual se abonará el importe total
3. La reserva no se puede anular, pero permite cambiar fechas
4. La reserva se puede anular con una multa que consiste en un porcentaje sobre el coste total.

Y con la *tupla* podría asociar cada reserva a su política de cancelación correspondiente.

Otra alternativa, en el caso de no querer usar *tuplas*, sería crear solo el *enum* y que este fuese un campo más de la reserva en el caso de que estas sean *structs*.

### Ejercicio 3: (20%)

Los clientes que se alojan en un hotel durante su estancia van usando diferentes servicios que ofrece el establecimiento (minibar, cafetería, peluquería, etc). Algunos de estos servicios no están incluidos en el precio correspondiente a su régimen de estancia y se deben facturar aparte. Por ejemplo, un cliente alojado en régimen de “Dormir con desayuno incluido” puede que cene un día en el hotel. Otro cliente, aun estando en régimen de “pensión completa” podría usar el servicio de la lavandería en algún momento.

Todos los servicios consumidos se registran mediante tickets que se van depositando uno encima del otro a medida que se van registrando. Al final del día, todos los tickets de un mismo servicio (del bar, por ejemplo), se pasan a recepción donde se clasifican según el cliente de cara a poder preparar la facturación total de la estancia de dicho cliente cuando haga el check-out.

Razonad qué TAD modeliza mejor la organización de los tickets que se acumulan en los diferentes servicios del hotel.

## **SOLUCIÓN**

Si nos ceñimos a la teoría asociada a esta PEC, deberíamos utilizar un TAD de tipo pila, dado que los tickets se van a apilar poniendo el primero en llegar debajo del todo y los siguientes encima.

Así, nos encontramos con una casuística a resolver con un comportamiento LiFo (*Last in First out*).

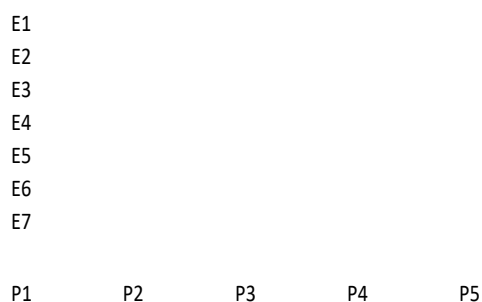
### Ejercicio 4: (40%)

Es habitual que los clientes que entran al hotel soliciten la posibilidad de dejar su equipaje en consigna mientras no pueden dejarlas en sus habitaciones, por haber llegado antes de que éstas estén disponibles. También puede ocurrir que un cliente deje el hotel y tenga necesidad de dejar las maletas unas horas mientras no continúa con su viaje.

Algunos hoteles tienen un espacio reducido para alojar dichos equipajes, de manera que no hay mucho espacio de maniobra: los equipajes se introducen en una habitación de forma que en el fondo queden las maletas que tardarán más en retirarse. Más cerca de la puerta de la consigna quedarán los equipajes que se retirarán antes. Ahora bien, aunque se intente alojar los equipajes con este criterio, a veces pasa que se necesita retirar un equipaje que no está accesible, provocando que se deban desempilar los primeros, retirar el equipaje deseado, y volver a empilar los que se habían apartado.

Este movimiento de equipajes se modeliza con las pilas mostradas a continuación, P1 a P5.

- La pila P1 representa la consigna del hotel.
- La pila P2 sirve para poder apilar las maletas que sea necesario quitar de en medio temporalmente para poder acceder a las maletas deseadas.
- Las pilas P3 a P5 son los carros donde los hoteles apilan las maletas de los clientes que abandonan el hotel. Estos carros sirven para llevar las maletas hasta un autocar o vehículo privado. Habrá en total 3 de dichos carros.



Asociadas al tipo `tStackCheckin` tenemos las instrucciones `push`, `top` y `pop` definidas como:

**action** `push` (**inout** `s`: `tStackCheckin`, **in** `p`: `tCheckin`)

**action** `pop` (**inout** `s`: `tStackCheckin`)

**function** `top` (`s`: `tStackCheckin`): `tCheckin`

```
push (P4, top (P1));  
pop (P1);
```

**Apartado 4.a (35%):** Dibujar la posición final de las pilas de las maletas después de moverlas ejecutando el siguiente conjunto de instrucciones:

## SOLUCIÓN

				E7
	E6			E4
	E1	E3	E5	E2
P1	P2	P3	P4	P5

**Apartado 4.b (65%):** Partiendo del resultado del apartado anterior, se pide que escribáis el conjunto de instrucciones necesarias para generar la secuencia de movimiento de equipajes que ubica los equipajes E1 y E3 en la pila temporal P2, los equipajes E5 y E2 en el carro P3, el equipaje E4 en el carro P5 y los equipajes E7 y E6 en el carro P4. La situación final debe ser la descrita en la imagen siguiente

	E3	E2	E6	
	E1	E5	E7	E4
P1	P2	P3	P4	P5

## SOLUCIÓN

```

push(P1, top(P2));
pop(P2);
push(P2, top(P3));
pop(P3);
push(P3, top(P4));
pop(P4);
push(P4, top(P5));
pop(P5);
push(P4, top(P1));
pop(P1);
push(P1, top(P5));
pop(P5);
push(P3, top(P5));
pop(P5);
push(P5, top(P1));
pop(P1);

```

## Criterios de corrección:

- Que se siga la notación algorítmica utilizada en la asignatura. Véase documento *Nomenclator* en la XWiki de contenido.
- Que se sigan las instrucciones dadas y se justifique adecuadamente la solución planteada.