

P1 PROBLEMA

- Gestión de la despensa
 - Stock de ingredientes = TAD Lista
 - Nombre
 - Cantidad
 - Alérgenos (lácteos, gluten)

P2 PROBLEMA

- Gestión de la despensa
- Gestión de la carta de postres
 - Lista de postres
 - Nombre
 - Precio
 - Receta (lista de ingredientes)

TADs

- Gestión carta de postres y stock de ingredientes -> prog. principal
 - Entrada acciones -> fichero
 - Salida -> consola modo texto
-
- **DessertList**: lista estática ordenada por nombre (campo *nDessert*)
 - **IngredientList**: lista dinámica con simple enlace y sin orden
 - **Request Queue**: Cola dinámica

PETICIONES

- Fase I
 1. Leer línea de fichero
 2. Parsear línea
 3. Insertar en cola
- Fase II
 1. Obtener la petición de frente de cola
 2. Procesar petición
 3. Eliminar frente de cola

Request Queue - Cola dinámica

CALENDARIO

MARZO						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
27	28	29	30	31	1	2
	INICIO P2					

ABRIL						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
					1	2
3	4	5	6	7	8	9
		CONTROL 1				
10	11	12	13	14	15	16
		NO LECTIVO				
17	18	19	20	21	22	23
		CONTROL 2				
24	25	26	27	28	29	30

MAYO						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
1	2	3	4	5	6	7
				ENTREGA P2		

P2 - Ufff... por donde empezar??? (I)

Control 1 – Del 3 al 7 de abril

- (a) Implementar TAD RequestQueue
 - (a.1) Buscar apuntes de clase
 - (a.2) Revisar las especificaciones
 - (a.3) Crear la Unit RequestQueue

```
unit RequestQueue;  
interface  
    type /* declaracion de tipos */  
        tCode = char;  
  
    /* cabeceras funciones/proc */  
    procedure A(param: tCode) ;  
  
implementation  
/* implementacion funciones /proc */  
procedure A(param: tCode) ;  
begin  
end;  
end.
```

P2 - Ufff... por donde empezar??? (I)

Control 1 – Del 3 al 7 de abril

- (a) Implementar TAD RequestQueue
 - (a.1) Buscar apuntes de clase
 - (a.2) Revisar las especificaciones
 - (a.3) Crear la Unit RequestQueue
- (b) Crear un programa de prueba
 - (b.1) Probar todas las funciones
 - (b.2) Probar casos típicos

```
program Test;  
uses RequestQueue;  
var  
    op: tCode;  
begin  
    op := 'C' ; A(op) ;  
    op := 'T' ; A(op)  
end.
```

P2 - Ufff... por donde seguimos???

Control 2 – Del 17 al 21 de abril

(a) Implementar TAD DessertList

(a.1) Buscar apuntes de clase

(a.2) Revisar las especificaciones

(a.3) Crear la Unit DessertList

```
unit DessertList;
interface

uses IngredientList;
type /* declaracion de tipos */
    tnDessert:string;
/* cabeceras funciones/proc */
procedure newDessert(param:tnDessert,...) ;
...

implementation
/* implementacion funciones/proc */
procedure newDessert(param: tnDessert,...) ;
begin
end;
end.
```


P2 - Ufff... por donde seguimos???

Control 2 – Del 17 al 21 de abril

(a) Implementar TAD DessertList

(a.1) Buscar apuntes de clase

(a.2) Revisar las especificaciones

(a.3) Crear la Unit DessertList

(b) Crear un programa de prueba

(b.1) Probar todas las funciones

(b.2) Probar casos típicos

```
program Test;  
uses DessertList, IngredientList;  
var  
    name: tnDessert;  
begin  
    name := "ChocoChake";  
    D (name) ;  
end.
```

Checklist

PRINCIPAL

- No contiene VARIABLES GLOBALES
- Está estructurado en funciones/procedimientos
- No contiene detalles de implementación de los TAD:
 - No ^
 - No indexación de arrays []
- Procesa correctamente el fichero de operaciones mínimo
- Al terminar la ejecución se elimina elemento a elemento las listas creadas
- El código está convenientemente comentado
- Se imprimen por pantalla los mensajes adecuados por cada operación

ESPECIFICACIÓN

Adjuntar PDF con la especificación del principal