INTRODUCCIÓN

En este documento de proyecto de programación abordamos la primera parte del proceso de creación del producto software. Se explicará el problema planteado y la solución que se le pretende dar. Con más extensión nos detendremos en la especificación de los requerimientos del sistema, y en todos los aspectos relativos a las fases de análisis y diseño.

Nuestro objetivo es realizar una aplicación que ayude en la gestión y automatización de la toma de decisiones en el ámbito de una cocina de restaurante. ¿Por qué mejora la gestión? Porque ayuda al jefe de cocina a realizar los menús según criterios de selección. ¿Por qué automatiza? Porque el proceso de creación de menús pasa a ser realizado con el soporte de la nueva herramienta software que los genera a partir de los criterios que el responsable de cocina considere oportunos, que en lugar de tener que estructurar diariamente durante un tiempo prolongado el menú que se va a servir, podrá con nuestro software hacerlo en unos instantes.

Desde el punto de vista económico el gerente de restaurante ejercerá un control más exhaustivo sobre los costes de cocina, ya que podrá por una parte planificar los menús a corto plazo, medio plazo o largo plazo a partir de los costes de producción de dichos menús, y por otra parte eligiendo aquellos que incluyen productos de temporada, que habitualmente se encuentran a mejor precio en los mercados. La herramienta software poseerá la capacidad de realizar menús de tipos específicos dirigidos a comensales con limitaciones en su alimentación, como vegetarianos, hipertensos u otros que se definen más adelante, con lo que desde el punto de vista comercial sirve de apoyo en la apertura del rango de clientes del negocio.

En restaurantes que habitualmente sirven menús, queremos informatizar con esta aplicación aquellos aspectos de la toma de decisiones del jefe de cocina que requieren un esfuerzo en tiempo y un coste económico derivado de la falta de planificación. Es decir, informatizar la creación de menús. Además, aprovecharemos el potencial de la automatización de este proceso ampliando la estrategia comercial facilitando la inclusión de nuevos platos en los menús para personas con necesidades especiales, incluyendo herramientas de control del coste de producción de los menús que ayuden a planificar en distintos plazos de tiempo y sirviendo de almacén de recetas y tipos de alimentos y otros que se podrán recuperar en cualquier momento, o bien actualizar o borrar a petición de los interesados.

En previsión podríamos incorporarlo en futuras ampliaciones del producto a otras herramientas de gestión de la empresa, como las que realizan contabilidad o controlan el stock, compartiendo datos de costos de producción o manteniendo actualizado el inventario de materias primas existente en la cocina, lo que podría ayudar a hacer más eficientes las compras. Además se podrían implementar herramientas gráficas y estadísticas que hagan comparativas entre los menús generados en distintos momentos de tiempo tomando los distintos aspectos de los que se componen.

Además, más allá del mundo de la hostelería, en el ámbito del usuario individual, esta aplicación podría implementarse en los nuevos dispositivos móviles, o bien como aplicación Web, convirtiéndose en una herramienta útil para cualquier persona que desee disponer al instante de un menú a su medida junto a las recetas para llevarlo a cabo.

Especificación de requerimientos del sistema.

El propósito de esta sección es proporcionar una visión completa y no ambigua de las funcionalidades y restricciones que debe reunir el producto para que sea aceptado por el cliente y servir de referencia y guía al equipo de desarrollo.

Está dirigida a:

Gerentes de restaurante

Jefes de Cocina

Usuarios individuales de aplicaciones móviles o Web

Analistas

DEFINICIONES

Menú: Conjunto de platos de constituye una comida. Está compuesto de primer plato, segundo plato y postre.

Pirámide alimentaria: La pirámide alimentaria es una guía de lo que se debe consumir diariamente para obtener los nutrientes que el cuerpo necesita. Para su interpretación se entiende que los alimentos dispuestos en el vértice superior son los que deben consumirse en menor cantidad y los que están cerca de la base son los que se deben comer con mayor frecuencia y en cantidades mayores, incluyendo las calorías que aportan. En la base de la pirámide están los alimentos que se deben consumir diariamente. Ahí están las patatas, cereales y sus derivados, verduras, hortalizas, frutas, leche y sus derivados, y aceite de oliva. Después, aparecen los que deben tomarse alternativamente varias veces a la semana, son las legumbres, frutos secos, pescados, huevos y carnes magras. Por último, en la cúspide, se encuentran los alimentos que sólo hay que comer de forma ocasional, concretamente carnes grasas, pastelería, bollería, azúcares y bebidas refrescantes.

Duración: Semanal (7 menús) o Mensual (31 menús)

Receta: Nota que comprende aquello de que debe componerse algo, y el modo de hacerlo.

Producto: Alimento o sustancia comestible que se usa para la elaboración de un plato.

Tipo de menú:

- Normal

Sin restricciones específicas

- Vegetariano

Sin carne ni pescado

- Diabéticos

Deben evitar:
Lácteos: todos los enteros como leche, nata, flanes, quesos, duros sobre todo.
Carnes: cerdo y sus derivados, embutidos, hamburguesas, vísceras, salchichas, conservas, chacinados y ahumados.
Verduras y hortalizas: que estén fritas como en la tempura.

Frutas: frutas conservadas en almíbar.
Cereales: todo la pastelería y bollería industrial, como tortas, pasteles, croissant, donuts, magdalenas.
Aceites y grasas: mantequilla, tocino, aceite de palma y aceite de coco.
Frutas secas: maníes salados, confitados y coco.
Bebidas: refrescos carbonatados, café irlandés, bebidas alcohólicas, bebidas con chocolate.
Especias y salsas: salsas hechas con mantequilla, margarinas o grasa animal.

- Hipertensos

Sin sal ni grasas saturadas o colesterol. Aumentar consumo de potasio (frutas y verduras)

- Bajo en calorías

Prioridad para aquellos alimentos bajos en calorías

PERSPECTIVA DEL PRODUCTO

Estará alojado en una estación de trabajo situada en el restaurante, en el área de cocina, y serán necesarias funciones de mantenimiento como la realización de copias de seguridad. El usuario tendrá acceso al puesto informático donde podrá hacer uso de la aplicación sin restricciones.

El producto generará menús a petición del usuario según los requisitos introducidos previamente. Además hará uso de ficheros para conservar la información introducida referente a recetas, alimentos y otras que más adelante se detallarán.

En el caso del uso en dispositivos móviles la aplicación estará alojada en el dispositivo final. Podría pensarse en posteriores desarrollos la implementación de la aplicación en Internet desde la Web, junto con la incorporación de una interfaz gráfica y un diseño adecuado que la haga más amigable y comercial.

Aunque el programa incorpora la función de añadir nuevas recetas y productos, para que sea funcional desde el primer momento será conveniente entregarlo precargado con al menos un número suficiente de recetas, que estimamos en 500 y sus productos correspondientes.

FUNCIONES

Función 1: Generación de menús a partir de requisitos

El usuario seleccionará uno a uno entre las diferentes opciones aquella que se adecue al menú que desea. Posteriormente visualizará el menú en pantalla ordenado por días y orden de platos. Si no hubiese una solución que cumpla todos los requisitos con los datos disponibles, se ofrecerá una lo más aproximada posible.

Opciones:

- Duración: Semanal o Mensual.
- Tipo de menú: Normal, Vegetariano, Diabéticos, Hipertensos, Bajo en calorías.
- Productos a evitar: Se selecciona de una lista de alimentos aquellos que se desean evitar.

- Productos de temporada: Da prioridad a las recetas que incluyen ingredientes de temporada.

Una vez seleccionadas las distintas restricciones se genera y visualiza el menú en pantalla.

- Se dará la opción de guardar el resultado en un fichero para posteriores consultas.

Todos los submenús tendrán la opción de volver al menú anterior.

Función 2: Gestión de datos

Para que el software sea de utilidad se deben introducir previamente los datos necesarios para la generación de menús, es decir, las restricciones que representan cada criterio. Toda la información necesaria para aplicar las restricciones y hacer la selección estará contenida en ficheros cuyos registros podrán ser actualizados por el usuario según lo requiera. Los criterios podrán ser modificados si sus características cambian. Para ello se presentarán las siguientes opciones:

Introducir o modificar recetas

Se podrá introducir una receta nueva o modificar algún campo de una existente. Para cada receta nueva se requerirá: nombre, posición (primero, segundo o postre) productos (ingredientes a elegir de una lista o crearlo si no existe), cantidades (valor numérico), procedimiento (descripción detallada de la receta) y tipo de menú para el que es adecuada (normal, vegetariano, etc.) Se asignará automáticamente un identificador numérico único a cada receta nueva.

Para modificar una receta se mostrará un menú de búsqueda para encontrar la receta que se desea modificar. Una vez confirmado que esa es la receta, se iniciará la lectura de los nuevos datos. Al final se mostrará la receta modificada y se almacenarán los cambios.

Introducir o modificar productos

Se podrá introducir un producto nuevo o bien modificar un producto existente. Para cada producto nuevo se requerirá:

Nombre

Categoría [0.Cereales, 1.Panes y harinas, 2.Legumbres, 3.Tubérculos, 4.Frutas, 5.Verduras, 6.Carne Roja y Cerdo, 7.Pollo, 8.Pescado, 9.Leche, queso y derivados, 10.Grasas y aceites, 11.Especias, 12.Bebidas, 13.Otros]

Temporada [1.Enero, 2.Febrero, 3.Marzo, 4.Abril, 5.Mayo, 6.Junio, 7.Julio, 8.Agosto, 9.Septiembre, 10.Octubre, 11.Noviembre, 12.Diciembre, 13.Siempre]

Precio por unidad de medida

Se asignará automáticamente un identificador numérico único a cada producto nuevo.

Para modificar un producto se mostrará un menú de búsqueda para encontrar el producto que se desea modificar. Una vez confirmado que ese es el producto, se iniciará la lectura de los nuevos datos. Al final se mostrará el producto modificado y se almacenarán los cambios.

Funcion 3: Consultas

Se podrán visualizar recetas, productos y menús determinados. Para ello se desplegará un menú de opciones. Las búsquedas se realizarán por campos, solicitando al usuario que anote el dato que desea buscar y el campo por el que lo desea buscar. Ir a la sección "Para realizar una modificación se mostrarán las los productos y se seleccionará aquel que se desea modificar. Una vez confirmado el producto a modificar, se capturarán los nuevos datos." para ver los campos de búsqueda de cada caso.

INTERFACES DE HARDWARE Y SOFTWARE

No han sido definidas aún.

FORMACIÓN

Antes de la implantación del sistema, se impartirá un curso de formación a los empleados que vayan a hacer uso de la aplicación. En el caso del usuario individual contará con manual online o bien un manual incorporado en el programa. Se pretende que el usuario pueda hacer un uso intuitivo del programa sin explicación previa alguna.

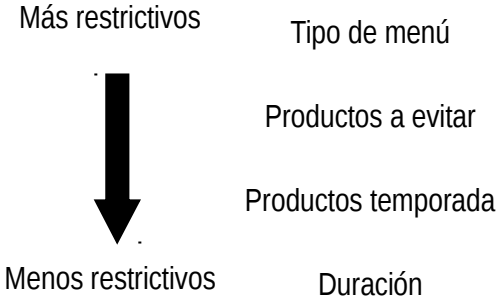
ANALISIS

Descripción del programa principal

El programa realiza dos funciones principales, por un lado visualizar en pantalla los menús generados a partir de los datos recogidos y por otro añade o modifica datos almacenados en ficheros. Las analizaré por separado.

Función 1: Generación de menús a partir de requisitos

El programa recopila información necesaria para la generación y visualización del menú. Después restringe la selección de recetas a las que cumplen los requisitos especificados, comenzando la misma desde los requisitos más restrictivos a los menos restrictivos.



A partir del archivo que contiene todas las recetas:

1. Seleccionar aquellas que corresponden al tipo de menú elegido.
2. Eliminar de la selección las que contienen productos a evitar.
3. Seleccionar o no aquellas que contienen productos de temporada.
4. Visualizar en pantalla el menú con un número de días conforme a la duración establecida.

Función 2: Gestión de datos

Desde esta función se podrá acceder a los ficheros de recetas y de productos para añadir nuevos registros o bien modificar los registros existentes. A partir del contenido de los distintos campos se realizará la selección de recetas que se lleva a cabo en la función de generación de menús, por lo que es fundamental su correcto tratamiento. Cuantos más datos almacenados en los ficheros de recetas y productos, más variedad se podrá ofrecer en la selección.

Se visualizará un menú (PANTALLA_B1) con las opciones existentes y se actuará en consecuencia.

INTRODUCIR O MODIFICAR RECETAS

Si se trata de una nueva receta se pedirán uno a uno todos los campos necesarios para completar el registro:

Código (se genera automáticamente)

Nombre

Posición (primer plato, segundo plato o postre)

Productos y cantidades (Ingrediente 1- Cantidad 1, Ingrediente 2- Cantidad 2…)

Procedimiento (Descripción detallada de la receta)

Tipo de Menú (para qué menú es apta: normal, vegetariano, hipertenso…)

Para realizar una modificación se mostrarán las recetas y se seleccionará aquella que se desea modificar. Una vez confirmada la receta a modificar, se capturarán los nuevos datos.

INTRODUCIR O MODIFICAR PRODUCTOS

Si se trata de un nuevo producto se pedirán uno a uno todos los campos necesarios para completar el registro:

Código (se genera automáticamente)

Nombre

Categoría (verdura, carne, pescado…)

Temporada (Meses que está en temporada)

Precio/unidad (euros/unidad)

Para realizar una modificación se mostrarán las los productos y se seleccionará aquel que se desea modificar. Una vez confirmado el producto a modificar, se capturarán los nuevos datos.

Definición de pantallas visualizadas en el programa

PANTALLA_A

PLANIFICADOR DE COCINA

1. Generar menús
2. Gestión de datos
3. Consultas

PANTALLA_B

PLANIFICADOR DE COCINA/Generar menú

1. Generar menú ahora
2. Anotar los criterios de selección

PANTALLA_B1

PLANIFICADOR DE COCINA/Gestión de datos

1. Introducir una receta
2. Modificar una receta
3. Introducir un producto
4. Modificar un producto

PANTALLA_B2

PLANIFICADOR DE COCINA/Consultas

1. Buscar recetas
2. Buscar productos
3. Buscar menús

PANTALLA_C1

PLANIFICADOR DE COCINA/Consultas/Buscar recetas

1. Mostrar todas
2. Buscar por nombre
3. Buscar por posición
4. Buscar por ingrediente
5. Buscar por tipo de menú

PANTALLA_C2

PLANIFICADOR DE COCINA/Consultas/Buscar productos");

1. Mostrar todos
2. Buscar por nombre
3. Buscar por categoria
5. Buscar por temporada

Entradas del programa principal

<table>
<tr><th colspan="4">Gestión de menús a partir de requisitos</th></tr>
<tr><td>Duración
(Un número)</td><td>Tipo
(Un número)</td><td>Productos a evitar
(varios tipos)</td><td>Productos temporada
(Un carácter)</td></tr>
<tr><td>Diario (1)</td><td>Normal (1)</td><td rowspan="6">Se pregunta si se desea evitar productos.
(S) o (N)
Si es (S) se capturan y validan los productos a evitar.
(una cadena de caracteres)
Se pregunta si se quiere elegir otro.
(S) o (N)</td><td>Si (S)</td></tr>
<tr><td>Semanal (2)</td><td>Vegetariano (2)</td><td>No (N)</td></tr>
<tr><td>Mensual (3)</td><td>Celíacos (3)</td><td rowspan="4">Para ‘S’ se pide el mes actual.
(Número de 1 a 12)</td></tr>
<tr><td></td><td>Diabéticos (4)</td></tr>
<tr><td></td><td>Hipertensos (5)</td></tr>
<tr><td></td><td>Bajo en Calorías (6)</td></tr>
</table>

<table>
<tr><th colspan="3">Consultas</th></tr>
<tr><td>Opción de menú
(un número)</td><td colspan="2">Campos de Búsqueda</td></tr>
<tr><td></td><td>Recetas</td><td>Productos</td></tr>
<tr><td>(1) Buscar recetas</td><td>Código (número)</td><td>Código (número)</td></tr>
<tr><td>(2) Buscar productos</td><td>Nombre (texto)</td><td>Nombre (texto)</td></tr>
<tr><td>(3) Buscar menús</td><td>Posición (1,2 o 3)</td><td>Categoría (numero)</td></tr>
<tr><td></td><td>Productos (texto)</td><td>Temporada (número)</td></tr>
<tr><td></td><td>Por tipo de menú (texto)</td><td></td></tr>
</table>

Gestión de datos							
Opción de menú (un número)	**Campos de registro** (varios tipos)		**Tipo de receta o producto** (texto)		**ID de receta o producto** (un número)	**Campo a modificar** (un número)	**Contenido del campo** (varios tipos)
(1) Nueva receta (2) Modificar receta (3) Nuevo producto (4) Modificar producto	**Recetas** Código (número) Nombre (texto) Productos (números) Cantidades (números) Unidades de medida (números) Procedimiento (texto)	**Productos** Código (número) Nombre (texto) Categoría [1] (numero) Temporada [2] (números) Precio (numero real) Unidad de medida (número)	**Recetas** Tipo plato (texto) (primero) Primer plato (segundo) Segundo plato (postre) Postre	**Productos** Tipo producto (texto)	Número identificativo de receta o producto	Un número correspondiente al campo de registro que se quiere modificar. Ver columna “Campos de registro”	Texto o número según campo. Ver columna “Campos de registro”

Salidas del programa principal

Función 1: Generación de menús según requisitos

Texto con el menú generado y lista de la compra.

Texto con recetas para su visualización.

Función 2: Gestión de datos

Esta función no tiene salidas.

Función 3: Consultas

Texto con lista de datos requeridos.

[1] 0.Cereales, 1.Panes y harinas, 2.Legumbres, 3.Tubérculos, 4.Frutas, 5.Verduras, 6.Carne Roja y Cerdo, 7.Pollo, 8.Pescado, 9.Leche, queso y derivados, 10.Grasas y aceites, 11.Especias, 12.Bebidas, 13.Otros

[2] 1.Enero, 2.Febrero, 3.Marzo, 4.Abril, 5.Mayo, 6.Junio, 7.Julio, 8.Agosto, 9.Septiembre, 10.Octubre, 11.Noviembre, 12.Diciembre, 13.Siempre

Solución general

Nivel 0. Programa principal.

//nombre del programa: Mproyecto.cpp

<INICIO>

Abrir fichero de recetas para leer
Contar número de registros de fichero recetas
Reservar memoria para array de recetas
Volcar fichero de recetas en array de recetas

Abrir fichero de productos para leer
Contar número de registros de fichero productos
Reservar memoria para array de productos
Volcar fichero de productos en array de productos

Abrir fichero de menús para leer
Contar número de registros de fichero menús
Reservar memoria para array de menús
Volcar fichero de menús en array de menús

Si las reservas de memoria no son correctas
    Gestionar error de apertura
    Salir del programa
Fin_si

repetir
  capturar y validar opcion_a
  si (opcion_a==ESC) //Salir
    volver_a = 1;
  sino
    según(opcion_a)
      para opcion_a==1: //generar menú
        limpiar pantalla
        Si hay recetas
            generar_menu
        sino
            gestionar error
        fin_si
      para opcion_a==2: //gestión de datos
        repetir
          capturar y validar opcion_b1
          si (opcion_b1==ESC) //volver
            volver_b1 = 1;
          sino

```
        segun (opcion_b1)
                para opcion_b1==1:          //introducir nueva receta
                  new_receta
                  ordenar_recetas
                para opcion_b1==2:          //modificar receta
                  Si hay recetas
                    mod_receta
                    ordenar_recetas
                  sino
                     gestionar error
                  fin_si
                para opcion_b1==3:          //introducir nuevo producto
                  new_producto
                  ordenar_productos
                para opcion_b1==4:          //modificar producto
                  Si hay productos
                     mod_producto
                     ordenar_productos
                  sino
                     gestionar error
                  fin_si
          fin_segun
       fin_si
   mientras (volver_b1 = 0)
para opcion_a=3:  //consultas
   repetir
     capturar y validar opcion_b2
     si(opcion_b2==ESC)  //si ha elegido volver
       volver_b2 = 1
     sino
       segun (opcion_b2)
         para opcion_b2==1:          //buscar recetas
           repetir
             capturar y validar opcion_c1
             si (opcion_c1==ESC)  //si ha elegido volver
               volver_c1 = 1
             sino
               Si no hay recetas
                     Gestionar error
               sino
                        segun (opcion_c1):
                          para  opcion_c1==1:
                              mostrar todas las recetas
                          para  opcion_c1==2:
```

```
                                    buscar_recetas //pasarle opcion_c1
                                para  opcion_c1==3:
                                    buscar_recetas //pasarle opcion_c1
                                para  opcion_c1==4:
                                    buscar_recetas //pasarle opcion_c1
                                para  opcion_c1==5:
                                    buscar_recetas //pasarle opcion_c1
                            fin_segun
                        fin_si
                    fin_si
                mientras (volver_c1 == 0)
            para opcion_b2==2:          //buscar productos
              repetir
                capturar y validar opcion_c2
                si (opcion_c2==ESC)   //volver
                  volver_c2 = 1
                sino
                        Si no hay productos
                                Gestionar error
                        sino
                            segun (opcion_c2):
                              para opcion_c2==1:
                                  mostrar todos los productos
                              para opcion_c2==2:
                                  buscar_productos //pasarle opcion_c2
                              para opcion_c2==3:
                                  buscar_productos //pasarle opcion_c2
                              para opcion_c2==4:
                                  buscar_productos //pasarle opcion_c2
                            fin_segun
                        fin_si
                fin_si
                mientras (volver_c2 == 0)
                case 3:   //buscar menús almacenados
                    Si hay menus almacenados
                            buscar_menu
                    sino
                            Gestionar error
                    fin_si
        fin_si
      mientras (volver_b2 == 0  y volver_c1 == 1 y volver_c2 == 1)
    fin_segun
  fin_si
mientras (volver_a=0 y volver_b1=1 y volver_b2=1)
```

abrir fichero recetas para escribir
Si la apertura no es correcta
    Gestionar error
sino
    Volcar array de recetas en fichero recetas
Fin_si
Cerrar el fichero recetas

abrir fichero productos para escribir
Si la apertura no es correcta
    Gestionar error
sino
    Volcar array de productos en fichero productos
Fin_si
Cerrar el fichero productos

abrir fichero menus para escribir
Si la apertura no es correcta
    Gestionar error
sino
    Volcar array de menus en fichero menus
Fin_si
Cerrar el fichero menus

Liberar memoria reservada para los arrays recetas, menus y productos
<FIN>

Nivel 0. Módulos.

Definición de módulos de Nivel 0.

VISUALIZAR PANTALLAS
Consultar contenido en “Para realizar una modificación se mostrarán las los productos y se seleccionará aquel que se desea modificar. Una vez confirmado el producto a modificar, se capturarán los nuevos datos.”

- visualizar pantalla_a
- visualizar pantalla_b1
- visualizar pantalla_b2
- visualizar pantalla_c1
- visualizar pantalla_c2

OTROS

- capturar y validar opcion_a
  repetir
  limpiar_pantalla
  visualizar pantalla_a
  inicializar variable volver_a
  escribir(OPCI0N(ESC sale):)
  leer (opcion_a)
  mientras(opcion_a != 1 y opcion_a != 2 y opcion_a!= 3 y opcion_a!= ESC)

- capturar y validar opcion_b1
  repetir
  limpiar_pantalla
  visualizar pantalla_b1
  inicializar variable volver_b1
  escribir(OPCI0N(ESC sale):)
  leer (opcion_b1)
  mientras(opcion_b1 != 1 y opcion_b1 != 2 y opcion_b1!= 3 y opcion_b1!=4 y
  opcion_b1!= ESC)

- capturar y validar opcion_b2
  repetir
  limpiar_pantalla
  visualizar pantalla_b2
  inicializar variable volver_b2
  escribir(OPCI0N(ESC sale):)
  leer (opcion_b2)
  mientras(opcion_b2 != 1 y opcion_b2 != 2 y opcion_b2!= 3 y opcion_b2!= ESC)

- capturar y validar opcion_c1
  repetir
  limpiar_pantalla
  visualizar pantalla_c1
  inicializar variable volver_c1
  escribir(OPCI0N(ESC sale):)
  leer (opcion_c1)
  mientras(opcion_c1 != 1 y opcion_c1 != 2 y opcion_c1!= 3 y
  opcion_c1 ¡= 4 y opcion_c1 ¡= 5 y opcion_c1!= ESC)

- capturar y validar opcion_c2
  repetir
  limpiar_pantalla
  visualizar pantalla_c2
  inicializar variable volver_c2
  escribir(OPCI0N(ESC sale):)
  leer (opcion_c2)
  mientras(opcion_c2 != 1 y opcion_c2 != 2 y opcion_c2!= 3 y
  opcion_c2 != 4 y opcion_c2 ¡= ESC)

- generar_menu
  capturar y validar opcion_b

```
Si no desea salir
    Según opcion
        Para opcion=1
                Iniciar select a valores por defecto
        Para opcion=2
                leer_tselect
    Fin_segun
    Reservar memoria para Plista //Lista que recogerá las recetas seleccionadas.
    Si asignación de memoria incorrecta
        Gestionar error
    Fin_si
    Crear_lista //Plista
    //Seleccionar recetas que coinciden con tipo de menú seleccionado
    Inicializar a 0 las variables num_primeros, num_segundos y num_postres
    Para(i=0, mientras existan recetas, i=i+1)
        Para(j=0, mientras hay tipos de menú, j=j+1)
                Si(tipo de menú de receta = tipo de menú de select)
                        Insertar_elemento //insertar receta en Plista
                        Actualizar contador según posición
                        //num_primeros, num_segundos o num_postres
                        Salir del bucle
                Fin_si
        Fin_para
    Fin_para
    Si en select hay productos a evitar
        //Eliminar las recetas que los contienen de Plista
        Eliminar_elemento //elimina receta en Plista
    Fin_si

    Reservar memoria para arrays de recetas seleccionadas
        //primeros, segundos, postres
    Si la asignación de memoria es incorrecta
        Gestionar error
    Fin_si

    Repartir el contenido de la Plista entre los tres arrays
        recetas que son primer plato a array Primeros
                aumentar size_primeros
        recetas que son segundo plato a array Segundos
                aumentar size_segundos
        recetas que son postre al array postres
                aumentar size_postres

    Si (size_primeros =0 o size_segundos =0 o size_postres =0)
       //si no hay recetas suficientes
       Gestionar error
    Fin_si

    //Hacer el listado de recetas
    Reservar memoria para el array de menus
        //menus recogerá los menús que se formen con
        //las recetas de primeros, segundos y postres.
    Si la asignación de memoria es incorrecta
        Gestionar error
    Fin_si
```

```
        //cargar menus
        Para (i=0, mientras existan menus, incrementa i)
            Menus[i].primer_plato=primeros[GNA(size_primeros)]
        Fin_para
        Para (i=0, mientras existan menus, incrementa i)
            Menus[i].segundo_plato=segundos[GNA(size_segundos)]
        Fin_para
        Para (i=0, mientras existan menus, incrementa i)
            Menus[i].postre=postres[GNA(size_primeros)]
        Fin_para

        Si se quieren elegir recetas con productos de temporada
            Almacenar en productos_tem los que están de temporada
            //en la temporada elegida en select
            Si las recetas seleccionadas tienen productos de temporada
                    Incluirlas en menus
            Fin_si
        Fin_si

        Visualizar menus en pantalla
        Preguntar si se desea guardar menus en archivo
        Si respuesta='S'
            Capturar y validar nombre de archivo
            Crear archivo
        Fin_si

        //Visualizar lista de la compra
    Fin_si
    Liberar memoria de Plista, primeros, segundos y postres
```

- new_receta
  repetir
  limpiar_pantalla
  leer_receta
  limpiar_pantalla
  escribir_receta
  Incrementar índice de recetas //size_r=size_r +1
  Preguntar si se desea repetir
  Mientras otra='S'

- ordenar_recetas
  Se usará para ello la función de librería qsort
- mod_receta
  repetir
  capturar y validar opcion_c1
  si no desea salir
  buscar la receta que desea modificar
  visualizar la receta en pantalla
  Confirmar selección
  //leer la nueva receta en la misma posición del array que ocupa la buscada
  Leer_receta
  Preguntar si se desea modificar otra receta
  fin_si

mientras se desee modificar recetas

- new_producto
  repetir
  limpiar_pantalla
  leer_producto
  limpiar_pantalla
  escribir_producto
  Incrementar índice de produtos //size_p=size_p +1
  Preguntar si se desea repetir
  Mientras otra='S'

- ordenar_productos
  Se usará para ello la función de librería qsort

- mod_producto
  repetir
  capturar y validar opcion_c2
  si no desea salir
  buscar el producto que desea modificar
  visualizar el producto en pantalla
  Confirmar selección
  //leer el nuevo producto en la misma posición del array que ocupa el buscado
  Leer_producto
  Actualizar recetas que incluyen el producto modificado
  Preguntar si se desea modificar otro producto
  fin_si
  mientras se desee modificar productos

- mostrar todas las recetas
  limpiar_pantalla
  escribir(LISTA DE RECETAS)
  para(i=0, mientras haya recetas, incrementar i)
  escribir_receta
  fin_para

- buscar recetas
  según opcion_c1
  para opcion=1 //buscar por codigo
  para opcion=2 //buscar por nombre
  para opcion=3 //buscar por posición
  para opcion=4 //buscar por ingrediente
  para opcion=5 //buscar por tipo de menu
  fin_segun

- mostrar todos los productos
  limpiar_pantalla
  escribir(LISTA DE PRODUCTOS)
  para(i=0, mientras haya productos, incrementar i)
  escribir_producto
  fin_para

- buscar productos

según opcion_c2
para opcion=1 //buscar por codigo
para opcion=2 //buscar por nombre
para opcion=3 //buscar por categoria
para opcion=4 //buscar por temporada
fin_segun

- buscar_menu
Presentar menú de opciones
Si opcion=1
Visualizar nombres de menus almacenados
Fin_si
Capturar y validar nombre de menú a visualizar
Si el nombre existe
Abrir fichero correspondiente
Visualizar menú contenido en el fichero
Fin_si

Nivel 1. Módulos.

Definición de módulos de Nivel 1.

- capturar y validar opcion_b
repetir
limpiar_pantalla
visualizar pantalla_b
escribir(OPCI0N(ESC sale):)
leer (opcion_b)
mientras(opcion_b != 1 y opcion_b != 2 y opcion_b!= ESC)

- crear_lista
Asigna NULL a Plista

- eliminar_elemento
elemento_buscado=**buscar_elemento**
Si elemento_buscado = NULL
Gestionar_error
Sino
Buscar la posición del elemento
Si es el primero
enlazar nodo por el caso especial
Sino
enlazar nodo por el caso general
Finsi
Liberar memoria del nodo eliminado
Finsi

- insertar_elemento
Reservar memoria para el nuevo elemento
Si la asignación es incorrecta
Gestionar error
Sino

almacenar nuevo_nodo
Si **lista_vacia**
insertar nuevo_nodo
sino
insertar nuevo_nodo en lista no-vacía
Fin_si
Fin_si

- lista_vacia
  Si Plista=NULL
  la lista está vacia
  fin_si

- leer_receta
  //generar y validar código
  codigo=(GNA(2000)+1000)
  escribir(“Nombre de la receta: ")
  leer(nombre)
  repetir //anotar y validar tipo de plato
  escribir(Primero, segundo o postre: ")
  leer(posicion)
  comprobar si existe posicion
  si no existe
  escribir(“Anote un valor correcto...")
  mientras la anotación sea incorrecta
  capturar ingredientes
  capturar procedimiento
  capturar y validar tipo de menú
  capturar y validar tipo de receta

- escribir_receta

visualiza en pantalla el contenido de una receta

- leer_producto

//generar y validar código
codigo=(GNA(10000)+10000)
//capturar nombre del producto
escribir(“Nombre del producto: ")
leer(nombre)
//capturar y validar categoría del producto
//capturar y validar temporada de producto
//capturar precio y magnitud

- escribir_producto

visualizar en pantalla el contenido de producto

- leer_tselect

//capturar criterio tipo_menu
//capturar criterio productos a evitar
//capturar criterio productos de temporada
//capturar criterio de duracion

Prototipos de las funciones

//FUNCIONES QUE VISUALIZAN MENUS DE OPCIONES

```
void pantalla_a(void);
void pantalla_b(void);
void pantalla_b1(void);
void pantalla_b2(void);
void pantalla_c1(void);
void pantalla_c2(void);
void print_categorias(void);
void print_temporadas(void);
```

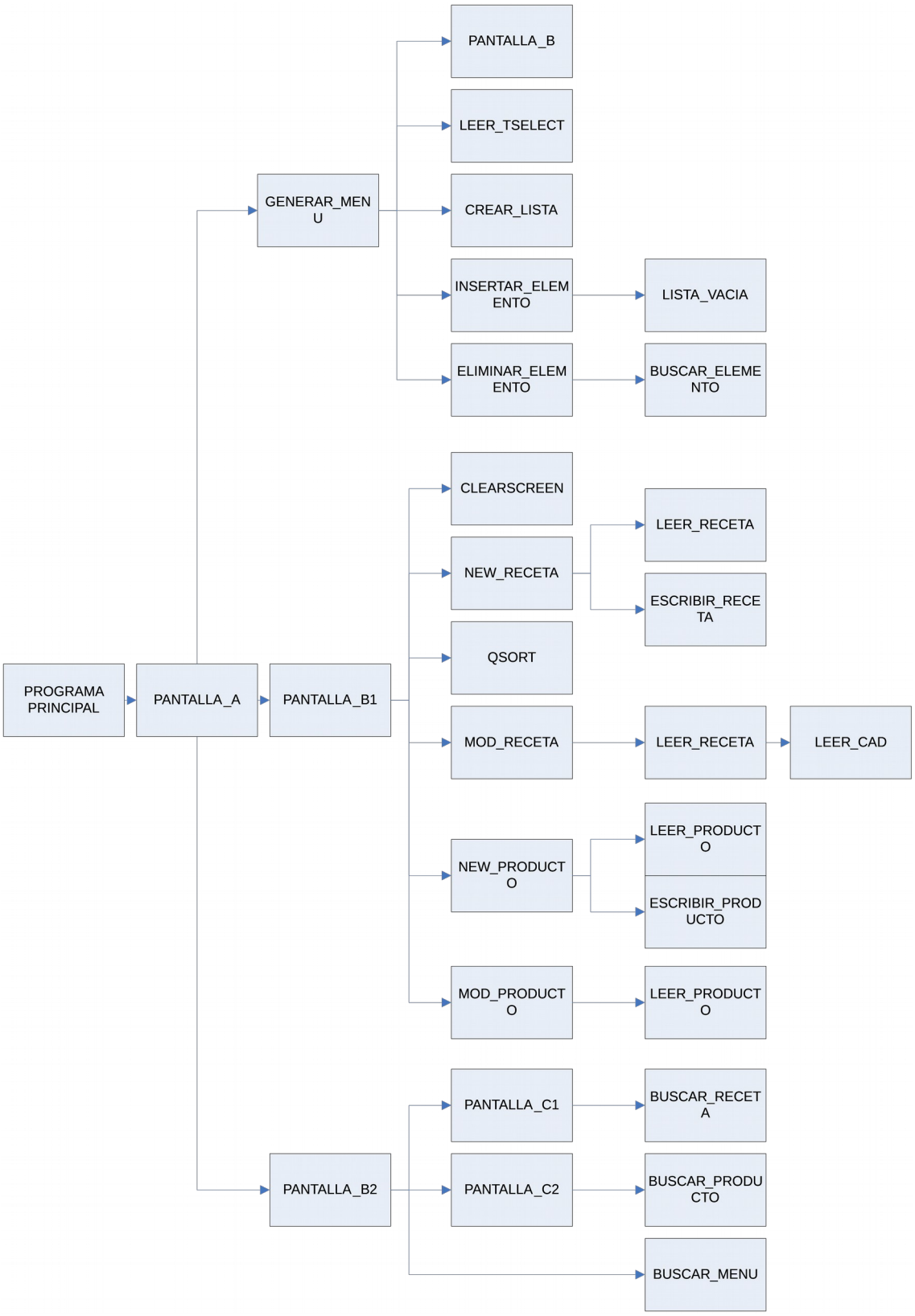
//FUNCIONES DEL PROGRAMA

```
void generar_menu(Treceta *, const int, Tproducto *, const int, cadena *, int &);
void new_receta(Treceta *, int &, Tproducto *, int &);
void mod_receta(Treceta *, int, Tproducto *, int &);
void new_producto(Tproducto *, int &);
void mod_producto(Tproducto *,int, Treceta *, int);
int buscar_receta(Treceta *, const int, const int);
int buscar_producto(Tproducto *, const int, const int);
void buscar_menu(cadena *, const int);
Treceta leer_receta(Treceta *, const int, Tproducto *, int &);
void escribir_receta(Treceta);
Tproducto leer_producto(Tproducto *, const int);
void escribir_producto(Tproducto);
void clearscreen();
int comp1(const void *, const void *);
int comp2(const void *, const void *);
void leer_tselect(Tselect &, Tproducto *, const int);
void LeerCad (char *);
```

//FUNCIONES DE LISTA ENLAZADA

```
void crear_lista(Tnodo * (&));
int lista_vacia(Tnodo *);
void insertar_elemento(Tnodo * (&), Treceta);
int eliminar_elemento(Tnodo * (&), int);
Tnodo * buscar_elemento(Tnodo *, int);
```

Diagrama de descomposición funcional



DISEÑO

Descripción de datos de las variables y estructuras

Estructuras

Tproducto_receta: Compuesta por una cadena de caracteres que almacena el nombre del ingrediente, un número real para la cantidad de ese ingrediente y otra cadena de caracteres que recoge la magnitud en la que se expresa dicha cantidad. Este tipo es usado al leer o modificar recetas, en un array de productos, que recoge los ingredientes que tiene la receta.

Definición de tipo:

```
typedef struct {        char producto[CPC];
                        float cantidad;
                        char magnitud[M];
                } Tproducto_receta;
```

TnumMagnitud: Compuesta de un número real que almacena el precio de un producto y una cadena de caracteres que almacena la magnitud en la que se expresa dicho precio. Es usada al leer o modificar productos.

Definición de tipo:

```
typedef struct {        float precio;
                        char magnitud[M];
                } TNumMagnitud;
```

Treceta: Usada al leer recetas, contiene todos los datos significativos de una receta de cocina. Un entero para el código que la identifica, una cadena para el nombre de la receta, una cadena para la posición que ocupa en el menú (primer plato, segundo o postre), una estructura Tproducto_receta para agrupar los ingredientes, una cadena de caracteres para almacenar el procedimiento de la receta y por último un array de cadenas para almacenar los tipos de menú para los que la receta es apta (vegetarianos, hipertensos…).

Definición de tipo:

```
typedef struct {        int codigo;
                        char nombre[NOMBRE];
                        char posicion[10];
                        Tproducto_receta datos_producto[FPC] ;
                        char procedimiento[PRO];
                        char tipo_menu[TM][LTM];
                } Treceta;
```

Tproducto: Usada al leer o modificar ingredientes, contiene un entero para el código que identifica al producto, una cadena para el nombre del ingrediente, una cadena para la categoría a la que pertenece, un array de cadenas para los meses en los que el producto está de temporada y un tipo TnumMagnitud para los datos de precio y magnitud.

Definición de tipo:

```
typedef struct {        int codigo;
                        char nombre[NOMBRE];
                        char categoria[30];
                        char temporada[12][12];
                        TNumMagnitud precio;
                } Tproducto;
```

Tselect: Estructura que almacena los datos necesarios para hacer la selección de recetas y componer el menú, según la elección del usuario. Contiene una cadena de caracteres que almacena el tipo de menú que se desea, un array de ingredientes para los productos que se quieren evitar, una cadena para indicar si se quiere dar prioridad a los productos de temporada y un entero que determina si se quiere componer un menú semanal o bien mensual.

Definición de tipo:

```
typedef struct {        char tipo_menu[LTM];
                        char evitar[10][CPC];
                        char temporada[LTM];
                        int duracion;
                }Tselect;
```

Tmenu: Estructura que almacena un menú compuesto de primer plato, segundo plato y postre. Contiene una estructura Treceta para cada uno. Se usa para recoger en un array del tamaño semanal (7 elementos) o mensual (31 elementos) las recetas seleccionadas.

Definición de tipo:

```
typedef struct {        Treceta primer_plato;
                        Treceta segundo_plato;
                        Treceta postre;
                }Tmenu;
```

Tnodo: Al generar menus se usa una lista auxiliar en la que se recogen inicialmente las recetas que cumplen el criterio de tipo de menú. Posteriormente se eliminan de ella las recetas que tienen productos a evitar.

Definición de tipo:

```
typedef struct Tnodo    {       Treceta info;
                                Tnodo * sig;
                        }Tnodo;
```

Cadena: Cadena de caracteres.

Definición de tipo:

```
typedef char cadena[CPC];
```

Variables del programa principal

//variables de tipo entero para control de pantallas:
opcion_a, volver_a
opcion_b1, volver_b1=1
opcion_b2,volver_b2=1
opcion_c1, volver_c1=1
opcion_c2, volver_c2=1

Estas variables se usan para moverse por las distintas pantallas que presenta el programa durante la ejecución, según el valor de opción_n se entra a una zona u otra y según el valor de volver_n se regresa o no al menú anterior más próximo.

//variables de tipo entero size_n
size_r, size_p, size_m hacen referencia al tamaño de los arrays de recetas, productos y menús que se cargan al inicio del programa procedentes de los archivos almacenados en memoria secundaria.

//índices
i, entero para el control de bucles.

```
//otras
  Treceta *recetas;
  Tproducto *productos;
  Arrays de recetas y productos.

  cadena * menus;
  Array de cadenas para guardar los nombres de los menús almacenados.

  FILE * Precetas, * Pproductos, *Pmenus;
  Punteros a tipo fichero para manejar los ficheros de recetas, productos y menús.

  char * ruta_menus="menus.txt";
  char * ruta_recetas="recetas.txt";
  char * ruta_productos="productos.txt";
  Cadenas de caracteres que almacenan el nombre de los ficheros que usa el programa.
```

Archivos utilizados. Descripción y estructura

menus.txt

Este archivo contiene cadenas de caracteres correspondientes a los nombres de los ficheros de menús generados por el usuario que se guardan en la carpeta menus_generados. Su información se usa para comprobar que el nombre elegido por el usuario no existe actualmente en la carpeta de archivos de menús generados.

Estructura: Registros de 100 caracteres cada uno.

recetas.txt

Este archivo contiene estructuras del tipo Treceta. Se usa para almacenar las recetas que usa el programa en memoria secundaria.

Estructura: Registros de tipo Treceta.

Ver “descripción de datos de estructuras y variables” en página 22.

productos.txt

Este archivo contiene estructuras del tipo Tproducto. Se usa para almacenar los productos que usa el programa en memoria secundaria.

Estructura: Registros de tipo Tproducto.

Ver “descripción de datos de estructuras y variables” en página 22.

menus_generados \ nombre_menu.txt

Dentro de la carpeta “menus_generados” se generan a discreción del usuario archivos con el nombre elegido y la extensión .txt, estos archivos guardan información sobre el menú generado por el programa para posteriores consultas.

Estructura: Registros del tipo Tmenu. 7 registros si el menú es semanal o 31 si el menú es mensual.

Ver “descripción de datos de estructuras y variables” en página 22.

Procesos realizados por el programa principal y los subprogramas

Programa principal

El programa principal realiza dos funciones principales. Por una parte trata la gestión de los archivos que se usan para extraer la información que se procesa durante la ejecución, lo que incluye abrirlos y cerrarlos, contar sus registros, reservar memoria para las estructuras en las que se vuelca su información y liberar la memoria antes de finalizar. Por otra parte es una estructura de menús que se presentan en pantalla y por las que el usuario se desplaza anotando opciones. A veces la elección de una de esas opciones implicará la llamada a algún subprograma.

Subprogramas

INTERFAZ DE PANTALLA_A

PROCESO: Presenta en pantalla el menú de opciones principal pantalla_a.

**********/

INTERFAZ DE PANTALLA_B

PROCESO: Presenta en pantalla el menú de opciones pantalla_b.

**********/

INTERFAZ DE PANTALLA_B1

PROCESO: Presenta en pantalla el menú de opciones pantalla_b1.

**********/

INTERFAZ DE PANTALLA_B2

PROCESO: Presenta por pantalla el menú de opciones pantalla_b2.

**********/

INTERFAZ DE PANTALLA_C1

PROCESO: Presenta por pantalla el menú de opciones pantalla_c1.

**********/

INTERFAZ DE PANTALLA_C2

PROCESO: Presenta por pantalla el menú de opciones pantalla_c2.

**********/

INTERFAZ DE PRINT_CATEGORIAS

PROCESO: Presenta por pantalla el menú de opciones print_categorias.

**********/

INTERFAZ DE PRINT_TEMPORADAS

PROCESO: Presenta por pantalla el menú de opciones print_temporadas.

**********/

INTERFAZ DE GENERAR_MENU

PROCESO: Selecciona recetas para confeccionar un menú, según un conjunto de criterios. Lo visualiza en pantalla y lo almacena e un archivo.

ENTRADAS: Punteros al primer elemento de los arrays de recetas, productos y nombres de menús generados. Enteros correspondientes al número de elementos de los dos primeros.

E/S: Entero por referencia correspondiente al número de elementos del array de menús generados.

PRECONDICIONES: Los arrays de productos y recetas deben contener datos.

**********/

INTERFAZ DE NEW_RECETA

PROCESO: Captura una receta y la añade al final del array de recetas. Se repite el proceso mientras se desee.

ENTRADAS: Punteros a los arrays de recetas y productos.

E/S: Enteros por referencia correspondientes al número de elementos de los arrays.

**********/

INTERFAZ DE MOD_RECETA

PROCESO: Busca la receta elegida, captura los nuevos datos y la asigna a la posición que le corresponde en el array de recetas. Se repite el proceso mientras se desee.

ENTRADAS: Punteros a los arrays de recetas y productos. Entero correspondiente al número de elementos del array de recetas.

E/S: Entero por referencia correspondiente al número de elementos del array de productos.

POSCONDICIONES: La receta elegida ha sido modificada.

**********/

INTERFAZ DE NEW_PRODUCTO

PROCESO: Captura un producto y lo añade al final del array de productos. Se repite el proceso mientras se desee.

ENTRADAS: Puntero al array de productos.

E/S: Entero por referencia correspondiente al número de elementos del array.

**********/

INTERFAZ DE MOD_PRODUCTO

PROCESO: Busca el producto elegido, captura los nuevos datos y lo asigna a la posición que le corresponde en el array de productos. Se repite el proceso mientras se desee.

ENTRADAS: Punteros a los arrays de recetas y productos. Enteros correspondientes al número de elementos de los arrays.

POSCONDICIONES: El producto elegido ha sido modificado.

**********/

INTERFAZ DE BUSCAR_RECETA

PROCESO: Busca y visualiza aquellas recetas que cumplen con el criterio elegido.

ENTRADAS: Entero de Opción (1- Buscar por código, 2- Buscar por nombre, 3- Buscar por posicion, 4- Buscar por ingrediente, 5- Buscar por tipo de menú)

SALIDAS: -1 si no la encuentra o la posición en el array de elemento encontrado. Si hay varios encontrados, devuelve -2.

PRECONDICIONES: El array de recetas contiene datos válidos.

POSCONDICIONES: Se devuelve un valor entero mayor o igual que -2.

**********/

INTERFAZ DE BUSCAR_PRODUCTO

PROCESO: Busca y visualiza aquellos productos que cumplen el criterio elegido.

ENTRADAS: Entero de Opción (1- Buscar por código, 2- Buscar por nombre, 3- Buscar por categoría, 4- Buscar por temporada)

SALIDAS: -1 si no la encuentra o la posición en el array de elemento encontrado.

PRECONDICIONES: El array de productos contiene datos válidos.

POSCONDICIONES: Se devuelve un valor entero mayor o igual que -1.

**********/

INTERFAZ DE BUSCAR_MENU

PROCESO: Busca y visualiza aquellos menús encontrados.

ENTRADAS: Puntero al array que recoge los nombres de los ficheros de menús almacenados y entero con su número de elementos.

**********/

INTERFAZ DE LEER_RECETA

PROCESO: Captura del teclado los datos de una estructura de tipo Treceta.

ENTRADAS: Punteros a arrays de recetas y productos, entero correspondiente al tamaño del array de recetas.

SALIDAS: Estructura de tipo Treceta.

E/S: Entero por referencia correspondiente al tamaño del array de productos.

POSCONDICIONES: La estructura devuelta contiene datos correctos.

**********/

INTERFAZ DE ESCRIBIR_RECETA

PROCESO: Visualiza en pantalla el contenido de una estructura de tipo Treceta

ENTRADAS: Tipo Treceta por valor.

PRECONDICIONES: La estructura está cargada con datos válidos.

**********/

INTERFAZ DE LEER_PRODUCTO

PROCESO: Captura del teclado los datos de una estructura de tipo Tproducto.

ENTRADAS: Puntero al array de productos y entero correspondiente a su tamaño.

SALIDAS: Estructura de tipo Tproducto.

**********/

INTERFAZ DE ESCRIBIR_PRODUCTO

PROCESO: Visualiza en pantalla el contenido de una estructura de tipo Tproducto

ENTRADAS: Tipo Tproducto por valor.

PRECONDICIONES: La estructura está cargada con datos válidos.

**********/

INTERFAZ DE CLEARSCREEN

PROCESO: Limpia toda la pantalla

**********/

INTERFAZ COMP1

PROCESO: Compara dos elementos de un array de estructuras Treceta por el campo código.

ENTRADAS: Punteros genéricos enviados por la función de librería Qsort.

SALIDAS: -1 si el primer elemento es menor que el segundo. 0 si el primer elemento es igual al segundo.

1 si el primer elemento es mayor que el segundo.

POSCONDICIONES: El valor devuelto es 0, -1 o 1.

**********/

INTERFAZ COMP2

PROCESO: Compara dos elementos de un array de estructuras Tproducto por el campo código.

ENTRADAS: Punteros genéricos enviados por la función de librería Qsort.

SALIDAS: -1 si el primer elemento es menor que el segundo. 0 si el primer elemento es igual al segundo.
1 si el primer elemento es mayor que el segundo.

POSCONDICIONES: El valor devuelto es 0, -1 o 1.

**********/

INTERFAZ LEER_TSELECT

PROCESO: Lee una estructura de tipo Tselect.

ENTRADAS: Puntero al array de productos y entero con su tamaño.

E/S: Una estructura de tipo Tselect por referencia.

**********/

INTERFAZ CREAR_LISTA

PROCESO: Inicializar la lista a estado vacío.

ENTRADAS: El puntero a la lista.

POSTCONDICIONES: La lista está creada y vacía(Plista apunta a NULL).

**********/

INTERFAZ LISTA_VACIA

PROCESO: Determinar si una lista está vacía o no. Una lista está vacía si su puntero apunta a NULL.

ENTRADAS: El puntero a la lista.

PRECONDICIONES: La lista debe estar creada.

SALIDAS: Un entero (0 si está vacía, -1 si no está vacía).

POSTCONDICIONES: Asociado al nombre de la funcion se devuelve 0 o -1.

**********/

INTERFAZ INSERTAR_ELEMENTO

PROCESO: Añadir un nuevo elemento a la lista, colocándolo en su sitio.

ENTRADAS: El nuevo elemento a añadir, el puntero a la lista.

PRECONDICIONES: La lista no debe estar llena (estática) o garantizar que hay memoria suficiente (dinámica).

POSTCONDICIONES: La lista queda con el nuevo elemento.

**********/

INTERFAZ ELIMINAR_ELEMENTO

PROCESO: Suprimir el elemento especificado de la lista.

ENTRADAS: El puntero a la lista y el elemento a suprimir.

PRECONDICIONES: La lista no debe estar vacía. El elemento a suprimir está en la lista y no se repite.

SALIDAS: Un entero con valor 0 si no ha encontrado el elemento a eliminar o -1 si lo ha encontrado.

POSTCONDICIONES: La lista queda con un elemento menos.

**********/

INTERFAZ BUSCAR_ELEMENTO

PROCESO: Buscar un elemento en una lista enlazada, ordenada, de enteros.

ENTRADAS: Puntero a primer elemento de la lista y entero con valor a buscar.

PRECONDICIONES: La lista no debe estar vacía.

SALIDAS: Puntero que apunta al nodo encontrado o NULL si no lo encuentra.

**********/

INTEFAZ LEER_CAD

PROCESO: Leer del teclado una cadena de caracteres.

ENTRADAS: Puntero a primer elemento de un array de caracteres.

POSTCONDICIONES: El puntero apunto a una cadena de caracteres recogida que termina con el carácter nulo.

**********/

Funcionamiento y uso del programa

Se trata de un programa muy sencillo que se puede utilizar correctamente de una forma intuitiva, ya que se han usado nombres muy descriptivos en los menús, para que el acceso sea claro.

El programa se ejecuta sin interfaz gráfica, por lo tanto solo dispone de texto sobre un fondo de color de relleno. Se nos presentan menús de los que debemos elegir una opción para continuar.

Al elegir esa opción bien se presentará un nuevo menú con más funciones o bien se ejecutará el proceso deseado, que podría incluir la inserción de datos por teclado. En estos casos se explica en pantalla cómo hacerlo.

La mayoría de las pantallas capturan la opción elegida con la mera pulsación, sin necesitar añadir Enter, además se usa la tecla de escape (esc) de la esquina superior izquierda del teclado para volver al menú anterior desde el actual o bien para salir del programa si estamos en el menú principal.

Al ejecutar el programa lo primero que debería visualizar es el siguiente menú:

PLANIFICADOR DE COCINA

1. Generar menús
2. Gestión de datos
3. Consultas

OPCI0N(ESC sale):

Pulse según su deseo la opción elegida.

Para planificar un menú semanal o mensual pulse 1.

Para gestionar los datos del programa (añadir o modificar recetas o ingredientes) pulse 2.

Para hacer consultas a los datos del programa (visualizar recetas, ingredientes o menús guardados con anterioridad, pulse 3.

Actúe de la misma forma en el resto de pantallas.

OTRAS CONSIDERACIONES

Descripción del desarrollo realizado

El desarrollo se ha realizado siguiendo el guión propuesto por el profesor de la asignatura y aplicando las estructuras y herramientas que se han estudiado hasta ahora y que se recogen en el libro “Programación en lenguajes estructurados” de Mª Asunción Criado Clavero.

Previo a lo dicho se eligió el objeto del proyecto de programación. En este caso particular seleccioné varios temas con los que tengo mayor afinidad y finalmente opté por este que me parecía más interesante y útil.

A partir de ese momento he seguido punto a punto las distintas fases del proceso que se definen en el índice de este documento.

Además de comenzar la escritura del código usando para ello el lenguaje de programación C standard y el compilador Borland C++ 5.02 en un ordenador equipado con Windows Vista.

Modificaciones realizadas a la solución inicial

2 de diciembre de 2010: Eliminada la opción de “limitar coste” con la que se desechan los resultados que superen un coste superior al indicado.

15 de marzo de 2011: Modificados los usuarios finales en “Introducción” y “Perspectiva del producto”.

1 de abril de 2011:

Modificado apartado “Añadir o modificar productos” de la sección “Funciones”

Modificado apartado “Formación”

Modificado Pseudocódigo para incluir control de acceso a pantallas

12 de abril de 2011: Añadida sección “Definición de pantallas visualizadas en el programa”

12 de abril de 2011: Modificado “Salidas del programa principal”

14 de mayo de 2011: Incorporada la función que crea la lista de la compra.

10 de junio de 2011: Refinado todo el documento para la entrega de la versión final.

Mejoras posibles del programa actual

Realización de una interfaz gráfica.

Inclusión de fotografías o vídeos de las recetas.

Mejora del algoritmo que realiza la selección de recetas.

Optar claramente por un tipo usuario final ya que no parece viable el desarrollo para un entorno profesional y un usuario particular en un mismo diseño. Me decanto por el usuario individual.

Incorporar más categorías de productos como: salsas, frutos secos, hongos o conservas.

Incorporar módulo para el cálculo de precios.

Permitir las búsquedas aproximadas.

Realizar copias de seguridad de los ficheros durante la ejecución.

INDICE

PROYECTO DE CREACIÓN DE SOFTWARE

Título: Menús a la carta.

Autor: Abraham Mesa.

1ºDAI

Versión: 1.2 Entrega final 13/06/2011