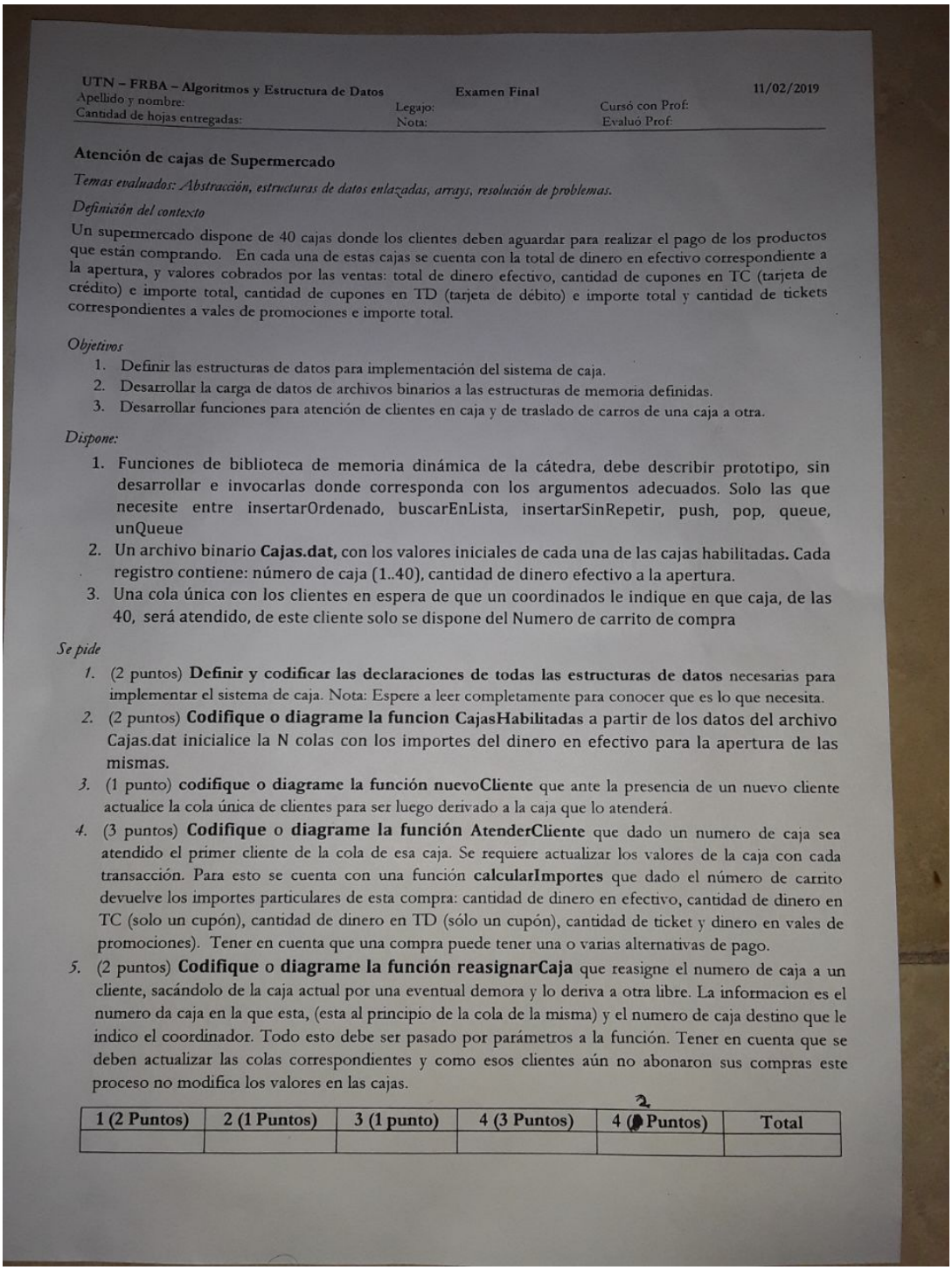
20191102

```
//40 cajas. Cada caja tiene un total de efectivo y valores cobrados por las ventas
// valorescobrados {total efectivo, cantidad de cupones en TC, importe total,
// cantidad de cupones en TD, importe total, canitdad de ticlets a vales, importe total}

1. Definir la estructuras de datos para el sistema de caja
2. Desarrollar la carga de datos de archivos binarios a las estructuras de memoria definidas
3. Desarrollar funciones para la atencion de clientes en caja y de translado de carros de una caja a otra.

Cajas.dat: valores iniciales de las cajas habilitadas.
Cada registro tiene un numero de caja y una cantidad de efectivo.

Cola unica con los clientes en espera que un coordinador le indique en que caja
será atendido.
Del cliente solo se sabe el numero de carrito de compra.

Se pide:
1. Definir y codificar las declaraciones de todas las estructuras de datos.
2. Funcion CajasHabilitadas apartir de los datos del archivo inicialice la N cajas con los
importes del dinero para la apertura
3. nuevoCliente que ante un nuevo cliente actualice la cola unica para luego ser derivado a la caja que lo atendera
4. AtenderCliente dado un numero de caja sea atendido el primer cliente de la cola de esa caja.
Hay que actualizar la caja con los valores de transacción.
La funcion calcularImportes con el numero de carrito devuelve los importes de la compra,
tener en cuent que una compra puede tener una o varias alternativas de pago.
5. reasignarCaja reasigna el numero de caja a un cliente, sacandolo de la caja actual.
Lo reasigna a un numero de caja. Ambos se pasan como parametro.
Se deben actualizar las colas correspondientes y como esos aun no abonaron sus compras
no modifica los valores en caja.
--------
1.
struct cajainfo
{float totalefectivo;
 int cantcuponesTC;
 float totalTC;
 int cantcuponesTD;
 float totalTD;
 int canttticketsavale;
 float totalticket;};

cajainfo[40];

struct NodoCola
{ int numcarrito;
  NodoCola* frente=NULL;
  NodoCola* fin=NULL;};

struct info
{ int numcaja;
  int totalefectivo;}

//Cajas.dat: Valores iniciales de las cajas habilitadas.
//f=fopen("Cajas.dat", "rb")

2. void CajasHabilitadas(FILE* f, cajainfo c[])
{ info c;
while(fread(&c, sizeof(info), 1, f)
{v[c.numcaja].totalefectivo=c.totalefectivo;}}

3. void nuevoCliente (NodoCola* &n, int numcarrito)
{queue(n->frente,n->fin, numcarrito);}

4. void atenderCliente (NodoCola *&cliente, int numcaja, cajainfo v[])
{ int numcarrito;
  cajainfo carrito;
  numcarrito=unqueue(cliente->frente, cliente->fin)
  carrito=calcularImportes (numcarrito);
  v[numcaja+1].totalefectivo= v[numcaja+1].totalefectivo+carrito.totalefectivo;
  v[numcaja+1].cantcuponesTC= v[numcaja+1].cantcuponesTC+carrito.cantcuponesTC;
```

```
  v[numcaja+1].totalTC= v[numcaja+1].totalTC+carrito.totalTC;
  v[numcaja+1].cantcuponesTD= v[numcaja+1].cantcuponesTD+carrito.cantcuponesTD;
  v[numcaja+1].totalTD= v[numcaja+1].totalTD+carrito.totalTD;
  v[numcaja+1].canttickетsavale= v[numcaja+1].canttickетsavale+carrito.canttickетsavale;
  v[numcaja+1].totalticket= v[numcaja+1].totalticket+carrito.totalticket;} //INCHEQUEBLE ESTO

reasignarCaja reasigna el numero de caja a un cliente, sacandolo de la caja actual.
Lo reasigna a un numero de caja. Ambos se pasan como parametro.
Se deben actualizar las colas correspondientes y como esos aun no abonaron sus compras
no modifica los valores en caja.
5. void reasignarCaja (int numcajaviejo, int numcajanuevo, cajainfo v[])
{ v[numcajanuevo-1]=v[numcajaviejo-1];}
```



UTN – FRBA – Algoritmos y Estructura de Datos **Examen Final** 11/02/2019

Apellido y nombre: Legajo: Cursó con Prof:

Cantidad de hojas entregadas: Nota: Evaluó Prof:

Atención de cajas de Supermercado

Temas evaluados: Abstracción, estructuras de datos enlazadas, arrays, resolución de problemas.

Definición del contexto

Un supermercado dispone de 40 cajas donde los clientes deben aguardar para realizar el pago de los productos que están comprando. En cada una de estas cajas se cuenta con la total de dinero en efectivo correspondiente a la apertura, y valores cobrados por las ventas: total de dinero efectivo, cantidad de cupones en TC (tarjeta de crédito) e importe total, cantidad de cupones en TD (tarjeta de débito) e importe total y cantidad de tickets correspondientes a vales de promociones e importe total.

Objetivos

1. Definir las estructuras de datos para implementación del sistema de caja.
2. Desarrollar la carga de datos de archivos binarios a las estructuras de memoria definidas.
3. Desarrollar funciones para atención de clientes en caja y de traslado de carros de una caja a otra.

Dispone:

1. Funciones de biblioteca de memoria dinámica de la cátedra, debe describir prototipo, sin desarrollar e invocarlas donde corresponda con los argumentos adecuados. Solo las que necesite entre insertarOrdenado, buscarEnLista, insertarSinRepetir, push, pop, queue, unQueue
2. Un archivo binario **Cajas.dat**, con los valores iniciales de cada una de las cajas habilitadas. Cada registro contiene: número de caja (1..40), cantidad de dinero efectivo a la apertura.
3. Una cola única con los clientes en espera de que un coordinados le indique en que caja, de las 40, será atendido, de este cliente solo se dispone del Numero de carrito de compra

Se pide

1. (2 puntos) **Definir y codificar las declaraciones de todas las estructuras de datos** necesarias para implementar el sistema de caja. Nota: Espere a leer completamente para conocer que es lo que necesita.
2. (2 puntos) **Codifique o diagrame la funcion CajasHabilitadas** a partir de los datos del archivo Cajas.dat inicialice la N colas con los importes del dinero en efectivo para la apertura de las mismas.
3. (1 punto) **codifique o diagrame la función nuevoCliente** que ante la presencia de un nuevo cliente actualice la cola única de clientes para ser luego derivado a la caja que lo atenderá.
4. (3 puntos) **Codifique o diagrame la función AtenderCliente** que dado un numero de caja sea atendido el primer cliente de la cola de esa caja. Se requiere actualizar los valores de la caja con cada transacción. Para esto se cuenta con una función **calcularImportes** que dado el número de carrito devuelve los importes particulares de esta compra: cantidad de dinero en efectivo, cantidad de dinero en TC (solo un cupón), cantidad de dinero en TD (sólo un cupón), cantidad de ticket y dinero en vales de promociones). Tener en cuenta que una compra puede tener una o varias alternativas de pago.
5. (2 puntos) **Codifique o diagrame la función reasignarCaja** que reasigne el numero de caja a un cliente, sacándolo de la caja actual por una eventual demora y lo deriva a otra libre. La informacion es el numero da caja en la que esta, (esta al principio de la cola de la misma) y el numero de caja destino que le indico el coordinador. Todo esto debe ser pasado por parámetros a la función. Tener en cuenta que se deben actualizar las colas correspondientes y como esos clientes aún no abonaron sus compras este proceso no modifica los valores en las cajas.

1 (2 Puntos)	2 (1 Puntos)	3 (1 punto)	4 (3 Puntos)	4 (2 Puntos)	Total