

<p align="center"><b>Universidad Tecnológica Nacional</b>  <b>Facultad Regional Avellaneda</b></p>			
<p align="center">Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos</p>			
<p>Materia: Laboratorio de computación I</p>			
<p>Apellido: Fecha:</p>			
Nombre:		Docentes <sup>(2)</sup> :	<b>Cardozo Marina, Bustamante Mathias</b>
División:	1° H	Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	<b>Primer Parcial Laboratorio I</b>		

(1) Las instancias válidas son: 1<sup>er</sup> Parcial (**PP**), Recuperatorio 1<sup>er</sup> Parcial (**RPP**), 2<sup>do</sup> Parcial (**SP**), Recuperatorio 2<sup>do</sup> Parcial (**RSP**), Final (**FIN**) . Marque con una cruz.

(2) Campos a ser completados por el docente.

**Desarrollar en ANSLC:**

Un sistema de ventas online desea gestionar la compra, venta y traslado de los productos, para ello se deberá realizar un programa de acuerdo a lo siguiente:

**Nota:** Establecer la relación entre Productos, Usuarios y Tracking (agregando los campos que crean necesarios), teniendo en cuenta que un producto tiene un único vendedor pero puede tener muchos compradores. Considerar que un tracking registra la venta entre un cliente y un vendedor.

```
typedef struct{
    int idProducto;
    short int isEmpty;           //Todos los estados que crean convenientes

    //AGREGAR RELACION
    char nombreProducto[25];
    float precio;
    short int categoria;        //Ej: Moda/Construccion/Tecnologia/etc
    int stock;                  //Cantidad de productos a vender
}Producto;

typedef struct{
    int idTracking;
    int idProducto;
    short int isEmpty;          //Todos los estados que crean convenientes

    //AGREGAR RELACIÓN
    int cantidad;               //Cantidad a comprar
    int distanciaKM;            //Distancia p/ entrega del producto
    long int horaLlegada;       //Calculo de 10seg cada 10km de distancia
}Tracking;

typedef struct{
    int idUsuario;
    short int isEmpty;          //Todos los estados que crean convenientes

    //AGREGAR RELACIÓN
    short int tipo;              //ADMIN - USUARIO
    char correo[25];             //Validado
    char password[10];           //Minimo 4 | Alfanumerico sin signo
    char direccion[50];          //Domicilio
    int codPostal;
}Usuario;
```

```
*****
** 1er EXAMEN LAB I - 1H **
*****
```

- 1) INGRESAR
- 2) REGISTRARSE

0) SALIR

1. **INGRESAR**: Se ingresara pidiendo usuario y contraseña, validando la existencia del mismo.
2. **REGISTRARSE**: ALTA DE USUARIO.  
VALIDANDO EL FORMATO DE TODOS SUS CAMPOS.

Luego del login será impreso el menú que corresponda al tipo de usuario

```
*****
** 1er EXAMEN LAB I - 1H **
***** USUARIO *****
*****
```

- 1) COMPRAR
- 2) VENDER
- 3) ESTADO DE COMPRAS
- 4) ESTADO DE VENTAS

0) SALIR

#### USUARIO

1. **COMPRAR**: Se listarán todos los productos disponibles para comprar ordenados por categoría. Luego se seleccionará el ID del producto a comprar y se le pedirá la cantidad a adquirir (se restará del stock), se informará al usuario el total de su compra y al confirmar **generará un ALTA DE TRACKING**.

Véase tabla de Código Postal para calcular distancia de entrega.

Para el campo *horaLlegada* será necesario emplear biblioteca <time.h>

CP - RANGO	KM
0001 - 1000	20
1001 - 3000	30
3001 - 5000	50
5001 - 8000	80
8001 - 9999	150

2. **VENDER**:  
ALTA DE PRODUCTO. Se listarán las categorías al momento de dar de alta.  
**REPONER STOCK**: Se listarán los productos que el usuario tenga en venta y seleccionando el ID podrá reponer stock del mismo.
3. **ESTADO DE COMPRAS**: Actualizar el estado de los productos cada vez que se consulte un tracking. En esta sección se listan todas las compras que haya realizado el usuario logueado con los datos que crean relevantes. Estos estados pueden ser EN VIAJE - ENTREGADO - CANCELADO. El usuario luego del listado, tiene la opción de SALIR o SELECCIONAR UN ID de un producto que esté EN VIAJE y cancelarlo, cambiando así su estado.
4. **ESTADO DE VENTAS**:
  - A. LISTADO DE VENTAS FINALIZADAS. Actualizar el estado de los productos cada vez que se consulte el tracking. En este listado estarán todas las ventas para el usuario logueado que hayan finalizado. Estados ENTREGADO - CANCELADO. (Si es que tiene productos en venta).
  - B. VER PRODUCTOS. Se listarán todos los productos que el usuario logueado aún tenga STOCK con toda su información (Si es que tiene productos en venta).

Luego del login será impreso el menú que corresponda al tipo de usuario

```
** 1er EXAMEN LAB I - 1H **
***** ADMIN *****
*****
```

## GESTION E INFORMES

- A) LISTAR ESTADO DE TODOS LOS USUARIOS
- B) LISTAR TODOS LOS PRODUCTOS POR CATEGORIA
- C) BAJA DE UN PRODUCTO
- D) BAJA DE UN USUARIO
- E) VER TRACKING GLOBAL
  
- 0) SALIR

ADMIN

### GESTIÓN E INFORMES:

- A. LISTAR ESTADO DE TODOS LOS USUARIOS: Listar ID - CORREO - ESTADO de todos los usuarios.
- B. LISTAR TODOS LOS PRODUCTOS POR CATEGORÍA: Listado de todos los productos dados de alta que se encuentran con stock en nuestro sistema. Ordenados por categoría y nombre.
- C. BAJA DE UN PRODUCTO: Se listan los productos activos en nuestro sistema y por medio del ID se dará de baja.
- D. BAJA DE UN USUARIO: Se listan los usuarios activos en nuestro sistema y por medio del ID se dará de baja.
- E. VER TRACKING GLOBAL: Se listará la siguiente información: ID de tracking - ESTADO en el que se encuentra - ID de comprador - ID de vendedor.
- F. FILTRAR POR NOMBRE DE PRODUCTO: Se ingresará el nombre a buscar y se listarán todos los productos que compartan el mismo nombre. Se deberá mostrar toda la info del producto ordenada por STOCK disponible.

### CALCULO DE HORA DE LLEGADA

El campo *distanciaKM* esta conformado por la tabla de CODIGO POSTAL. Ej: Si un usuario compra un producto y su domicilio tiene un **Código Postal 5500** le corresponde **80km DE DISTANCIA PARA LA ENTREGA**.  
Por cada 10 kilómetros de distancia, el producto se demora 20 segundos en llegar.  
Para seguir y modificar el estado del tracking se usarán las funciones brindadas a continuación.

```
long int time_Current(void){
    return time(NULL);
}

long int time_Add(double secondsAdd){
    return secondsAdd + time_Current();
}
```

Estas funciones darán la hora actual o la hora más el tiempo en segundos que sumemos. Cuando dice “*Actualizar el estado de los productos cada vez que se consulte un tracking.*” Quiere decir que tendrán una función que actualice los estados del tracking, realizándose el cálculo entre en la hora que se seteo en *horaLlegada* con la hora actual. Cada vez que se consulte el listado los tracking deben verse con sus estados actualizados, en el caso que esté EN VIAJE y al momento de listarlo ya pasó el tiempo de entrega deberá verse el estado ENTREGADO.

NOTAS:

- Se deberá desarrollar una biblioteca para todas las entidades, la cual contendrá todas las funciones.
- Las validaciones deberán estar en una biblioteca aparte.
- No se podrá ingresar a los casos en los que no haya nada que mostrar, ej: los listados; sin antes haber realizado al menos un alta de lo que se desea mostrar.
- Para la entidad Usuario, no hace falta realizar el ABM, sino que se podrá declarar e inicializar los datos de manera directa (hardcode).
- El código deberá tener comentarios con la documentación de cada una de las funciones y respetar las reglas de estilo de la cátedra.
- Realizar funciones modularizadas y reutilizables.
- El día del examen deberán traer este parcial funcionando (sin errores de compilación, de ejecución y sin warnings) para poder acceder a la instancia evaluatoria.

**Cualquier intento de copia anulará el parcial de todos los estudiantes involucrados en el plagio.**