

# UTN – FR Mar del Plata - Técnico Universitario en Programación

## Programación I

### Trabajo Práctico Adicional. Estructuras de Datos.

#### Algoritmos sobre arreglos de estructuras.

Sean las estructuras:

```
typedef struct{  
    int idProducto;  
    int stock;  
    float precio;  
    char nombre[DIMPALABRAS];  
    char categoriaProducto[DIMPALABRAS];  
}stProducto;
```

```
typedef struct{  
    char categoria[DIMPALABRAS];  
    stProducto productos[DIM];  
    int productosDisponibles; //funciona de validos  
}stCategorias;
```

1. Generar las funciones necesarias para la carga de un arreglo de tipo *stProductos* llamado **productosSinCategorizar** que contendrá todos los productos disponibles en una tienda. (Modularizar) Opcional: hacer el *idProducto* autoincremental.
2. Escribir una función que genere un arreglo de tipo *stCategorias* llamada **productosCategorizados** que guarde todas las categorías presentes en el arreglo de productos (sin que se repitan). Nota: inicializar el campo *productosDisponibles* en 0.
3. Escribir una función que recorra el arreglo **productosSinCategorizar** y vaya guardando cada producto en el arreglo productos contenido en cada categoría dentro de **productosCategorizados**. (muy importante modularizar)  
Ejemplo: si el producto de **productosSinCategorizar** pertenece a la categoría “Hogar” deberás buscar en **productosCategorizados** donde se encuentre dicha categoría y agregar en **productos** el producto, este deberá ser recorrido con el índice guardado en **productosDisponibles** que guarda los validos de ese arreglo.
4. Escribir una función que busque un producto por su ID en el arreglo de **productosCategorizados**.
5. Escribir una función que me permita actualizar el stock o el precio de un producto buscado por su ID.
6. Escribir una función que muestre todos los productos de una categoría recibida por parámetro.
7. Escribir una función que muestre por pantalla todos los productos (sin importar la categoría) para los cuales su stock sea menor a un número recibido por parámetro.
8. Escribir una función que busque en **productosCategorizados** el producto con el precio más alto o más bajo según la condición que recibe por parámetro y lo muestre anunciando si es el producto con el precio más alto o el más bajo.
9. Escribir una función que elimine de **productosCategorizados** todos los productos que no tiene stock, mostrar los productos que fueron eliminados.