

# LABORATORIO DE COMPUTACIÓN 2



MAPPLE ARG



## INDICE

<b>Introducción.....</b>	<b>3</b>
<b>Descripción del funcionamiento.....</b>	<b>3</b>
<b>Menú Principal:.....</b>	<b>3</b>
<b>Stock de Productos: .....</b>	<b>3</b>
<b>Gastos: .....</b>	<b>5</b>
<b>Presupuestos y Facturas: .....</b>	<b>6</b>
<b>Balances: .....</b>	<b>7</b>
<b>Diagrama de Clases.....</b>	<b>8</b>



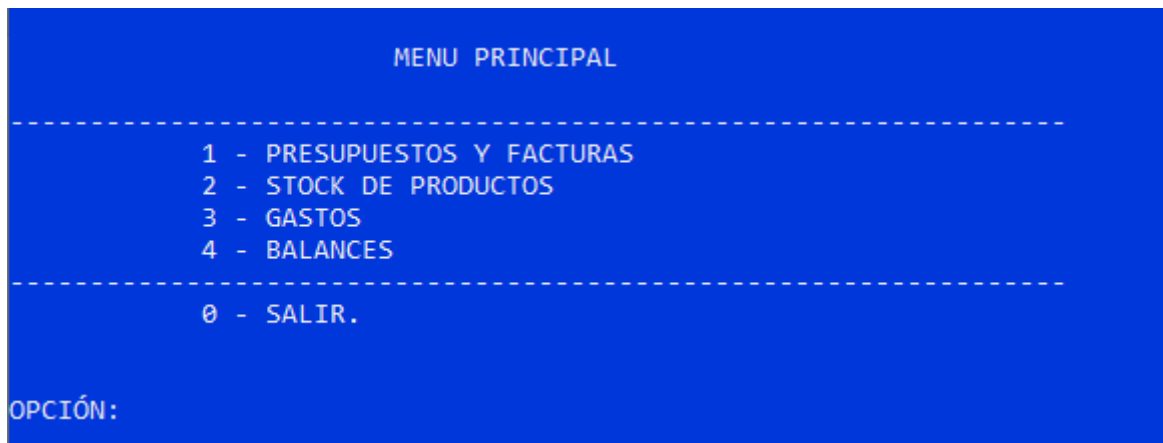
## Introducción

El equipo Mapple Arg, compuesto por Pedro y Diego, elaborará un sistema informático desarrollado íntegramente en C++, el cual permitirá gestionar stock, presupuestos, facturas y balances de una empresa familiar dedicada a la venta de revestimiento antihumedad, también compatible con otras empresas dedicadas a las ventas de insumos. Para esto contará con un menú principal y una serie de submenús que serán la interfaz gráfica para la utilización del comerciante.

## Descripción del funcionamiento

Menú Principal:

Desde él se pueden acceder a los diferentes módulos del programa.



Stock de Productos:

El sistema permitirá registrar nuevos productos en el submenú de **Stock**. Para esto se solicitará un código de producto (que actuará de identificador unívoco), una descripción y el precio de venta inicial.



```
CARGAR NUEVO PRODUCTO
-----
INGRESE CODIGO DEL PRODUCTO: 013
INGRESE DESCRIPCIÓN DEL PRODUCTO: Adhesivo x 20 Kg
INGRESE PRECIO DEL PRODUCTO: $5450
-----
PRODUCTO AGREGADO
```

Posteriormente se podrá actualizar el precio de venta de la mercancía.

```
ACTUALIZAR PRECIO PRODUCTO
-----
INGRESE CODIGO DE PRODUCTO A MODIFICAR: 011
PRECIO A MODIFICAR: $1600
INGRESE NUEVO PRECIO DEL PRODUCTO: $1700
-----
PRECIO ACTUALIZADO
```

Para registrar el ingreso de stock, se deberá registrar la compra en la opción correspondiente del módulo.

```
REGISTRAR COMPRA DE PRODUCTO
-----
INGRESE CODIGO DE PRODUCTO A CARGAR STOCK: 012
STOCK ACTUAL: 0
INGRESE UNIDADES COMPRADAS: 100
INGRESE PRECIO DE CADA UNIDAD: $2400
NUEVO STOCK: 100
-----
SE HA AGREGADO STOCK
```



Este "Gasto" se va a ver reflejado en el submenú de Gastos -> Ver gastos realizados.

VER GASTOS REALIZADOS					
ID	FECHA	DESCRIPCIÓN	PRECIO	UNIDADES	PRECIO TOTAL
1	3/11/21	11 - Adhesivo x 5 kg	\$1000.00	15	\$15000.00
2	3/11/21	12 - Adhesivo x 10 kg	\$2400.00	15	\$36000.00
3	3/11/21	12 - Adhesivo x 10 kg	\$2400.00	85	\$204000.00
4	3/11/21	13 - Adhesivo x 20 kg	\$4300.00	10	\$43000.00

Y por último tendrá una visualización de todos los productos con su stock y precio correspondiente.

***** STOCK DISPONIBLE *****				
CÓDIGO	DESCRIPCIÓN	PRECIO	STOCK	DISPONIBLE
13	Adhesivo x 20 kg	\$5450.00	10	
11	Adhesivo x 5 kg	\$1700.00	15	
12	Adhesivo x 10 kg	\$2900.00	100	

Gastos:

En el submenú de **Gastos**, se podrán cargar gastos varios como insumos de oficina, fletes contratados, etcétera.

```
CARGAR NUEVO GASTO
-----
INGRESE DESCRIPCIÓN DEL GASTO: Lapiceras
INGRESE PRECIO UNITARIO DEL GASTO: $6
INGRESE CANTIDAD DE UNIDADES: 10
-----
GASTO GRABADO
```



Tenemos otra opción en este submenú para acceder a un listado de todos los gastos realizados con sus respectivas fechas, descripciones, precios unitarios y precios totales.

VER GASTOS REALIZADOS						
ID	FECHA	DESCRIPCIÓN	PRECIO	UNIDADES	PRECIO TOTAL	
1	31/10/21	11 - Adhesivo x 4 Kg	\$1000.00	100	\$100000.00	
2	31/10/21	12 - Adhesivo x 10 Kg	\$2400.00	100	\$240000.00	
3	31/10/21	Lapiceras	\$6.00	10	\$60.00	

### Presupuestos y Facturas:

Para la elaboración de un **Presupuesto** el programa generará un código (ID) de presupuesto automáticamente y solicitará al usuario del programa los ítems a presupuestar.

NUEVO PRESUPUESTO		FECHA: 01/01/2021		
CÓDIGO	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	TOTAL
134	Revestimiento Antihumedad	20	505	10100
135	Revestimiento Antihumedad xm2	20	545	10900
F3 - AGREGAR    F5 - ELIMINAR    F7 - CERRAR    F9 - RECALCULAR    F11 - GENERAR FACTURA				
CÓDIGO: 136				
Revestimiento Antihumedad xm2    CANTIDAD:12				

Si se agrega un ítem de un producto ya ingresado, ambos se agruparán sumándose la cantidad de unidades vendidas. También se podrá eliminar un ítem de la lista identificándolo con su código de producto.

Por último, el presupuesto tendrá un plazo de validez (definido por la cantidad de días que decida el usuario, con un límite de 90 días) durante el cual se respetarán los precios guardados al momento de la generación de este. Pasada la fecha límite, el usuario podrá optar por conservar los precios pactados al generar el presupuesto, o actualizarlos a los



valores actuales. Si quisiera realizar una modificación del presupuesto, los valores deberán actualizarse.

Al guardar un nuevo presupuesto, se mostrará por pantalla el ID del mismo, el cual podrá ser usado posteriormente para volver a abrirlo, por ejemplo, para editarlo.

Si el comprador acepta, debemos generar la **Factura** correspondiente, mediante la opción disponible en la visualización del presupuesto. Esta tendrá un identificador propio generado automáticamente. El sistema solicitará los datos personales de los clientes. Estos se guardarán para posteriormente ser cargados con mayor facilidad (al ingresar el DNI).

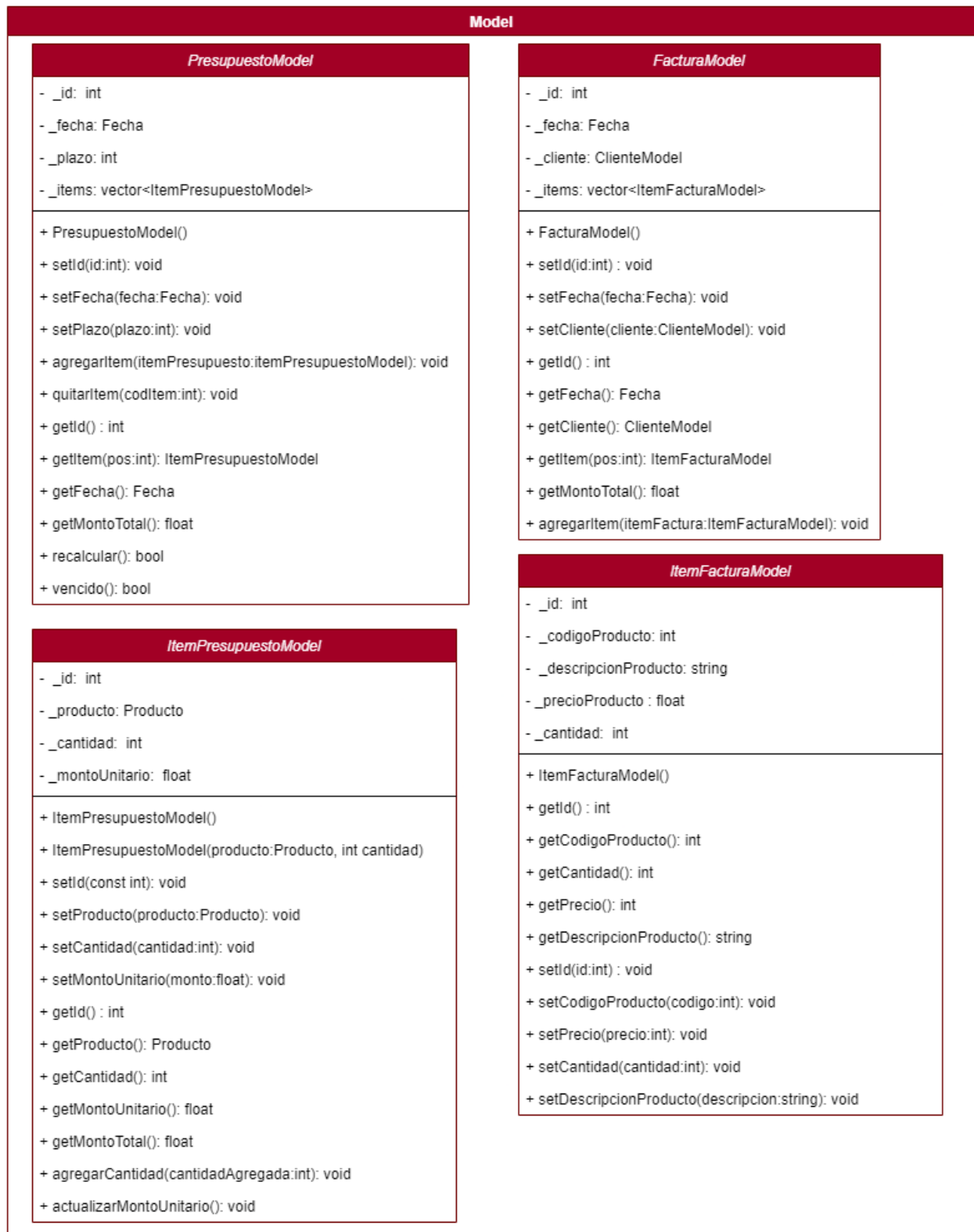
VER FACTURAS REALIZADAS				
-----				
INGRESAR ID FACTURA: 00254				
FECHA: 31/10/2021				
DNI COMPRADOR:32617997				
CODIGO	DESCRIPCION	PRECIO UN.	UNIDADES	PRECIO TOTAL
11	Adhesivo x 4 Kg	\$1700.00	1	\$1700.00
-----				
134	Revestimiento Antihumedad F134	\$500.00	10	\$5000.00
-----				
				PRECIO FINAL: \$6700.00

#### Balances:

Aquí se podrá hacer una exportación a un archivo con formato CSV con un resumen de los movimientos (según los gastos y facturas emitidas) de un mes seleccionado.



## Diagrama de Clases







#### *ClienteModel*

- \_DNI: int  
- \_nombre: string  
- \_apellido: string  
- \_direccion: string  
- \_telefono: string

+ ClienteModel()  
+ ClienteModel(DNI:int, nombre:string, apellido:string, telefono:string)  
+ getDNI(): int  
+ getDireccion(): string  
+ getNombre(): string  
+ getApellido(): string  
+ getTelefono(): string

#### *ProductoModel*

- \_codigoProducto: int  
- \_descripcionProducto: string  
- \_precioProducto: float  
- \_stockDisponible: int

+ ProductoModel()  
+ getCodigoProducto(): int  
+ getDescripcionProducto(): string  
+ getPrecioProducto(): float  
+ getStockDisponible(): int  
+ setCodigoProducto(codigo:int): void  
+ setDescripcionProducto(descripcion:string): void  
+ setPrecioProducto(precio:float): void  
+ agregarStock(stockAgregado:int): void  
+ quitarStock(stockQuitado:int): void

#### *GastoModel*

- \_id: int  
- \_montoUnitario: float  
- \_descripcionGasto: string  
- \_unidades: int  
- \_fecha: Fecha

+ GastoModel()  
+ getId(): int  
+ getMontoUnitario(): float  
+ getDescripcionGasto(): string  
+ getUnidades(): int  
+ getMontoTotal(): float  
+ getFecha(): Fecha  
+ setId(id:int): void  
+ setDescripcionGasto(descripcionGasto:string): void  
+ setFecha(fecha:Fecha): void  
+ setMontoUnitario(monto:float): void  
+ setUnidades(unidades:int): void

#### *Fecha*

- \_dia: int  
- \_mes: int  
- \_anio: int  
- diasDelMes(mes:int, anio:int)

+ Fecha()  
+ Fecha(dia:int,mes:int,anio:int)  
+ getDia(): int  
+ getMes(): int  
+ getAnio(): int  
+ getStrFecha(): string  
+ operator+(dias:int): Fecha  
+ operator<=(fecha2:Fecha): bool

