

# TP 2

Integrantes:

- Paciulli Tomas Agustin
- Fornes Ezequiel Christian
- Cantarella Luciano
- Semperena Marisol
- Mirando Lucas Abel
- Ramos Lautaro Agustin

link del repo

[https://github.com/Marsemperena/plat\\_desa\\_tp/tree/tp2-entrega](https://github.com/Marsemperena/plat_desa_tp/tree/tp2-entrega)

Para serializar los objetos se utilizo la librería Newtonsoft, que pasa las entidades a json, estos archivos se guardan por defecto en %appdata%/mercado, pero el usuario puede configurar libremente donde desea guardar los archivos mediante la pantalla de configuración. los dao levantan ese path y lo combinan con le nombre correspondiente para cada clase

ejemplo de usuario junto con el archivo generado

```
static string fileName = Path.Combine(LocalFileManager.userpath, "Usuario.json");
```

```

{
  "id": 0,
  "dni": 0,
  "nombre": "admin",
  "apellido": "admin",
  "mail": "admin@gmail.com",
  "password": "admin",
  "tipo": "admin",
  "cuilCuit": "000",
  "MiCarro": {
    "id": 0,
    "productos": {}
  }
},
{
  "id": 1,
  "dni": 1,
  "nombre": "cliente",
  "apellido": "cliente",
  "mail": "cliente@gmail.com",
  "password": "cliente",
  "tipo": "cliente",
  "cuilCuit": "001",
  "MiCarro": {
    "id": 0,
    "productos": {}
  }
},
{
  "id": 2,
  "dni": 0,
  "nombre": "lucas",
  "apellido": "miranda",
  "mail": "test",
  "password": "123",
  "tipo": "02",
  "cuilCuit": "cliente",
  "MiCarro": {
    "id": 0,
    "productos": {}
  }
},
{
  "id": 3,

```

para mantener la integridad de los datos en la clase mercado cada vez que se modifica una lista esta se impacta en su correspondiente archivo

```

public List<Categoria> categorias= new List<Categoria>();

public List<Producto> productos = new List<Producto>();

public List<Usuario> usuarios = new List<Usuario>();

public List<Compra> compras ;

    4referencias
    public Mercado()
    {
        .....

        this.productos = ProductoDAO.getAll();
        this.categorias = CategoriaDAO.getAll();
        this.usuarios = UsuarioDAO.getAll();
        this.compras = new List<Compra>();
    }

```

```

public bool agregarProducto (string nombre, double precio, int cantidad, int id_Categoria)
{
    int idActual = 0;
    foreach (Producto prod in productos)
    {
        if (prod.id > idActual) { idActual = prod.id; }
    }
    try
    {
        this.productos.Add(new Producto(idActual +1,nombre, precio, cantidad, this.categorias[id_Categoria]));
        ProductoDAO.saveAll(productos);
        return true;
    }
    catch (Exception e)
    {
        Console.WriteLine("Ocurrio un error al intentar dar de alta el producto, por favor intente nuevamente");
        return false;
    }
}

```

para pasar datos de los formularios al back estos pasan por validaciones para evitar pasar errores por tipos de datos

```

bool idOK = int.TryParse(tabla.Rows[e.RowIndex].Cells[0].Value.ToString(), out id);
bool precioOK = double.TryParse(tabla.Rows[e.RowIndex].Cells[2].Value.ToString(), out precioProd);
bool cantOK = int.TryParse(tabla.Rows[e.RowIndex].Cells[3].Value.ToString(), out cantProd);
bool idcatgOK = int.TryParse(tabla.Rows[e.RowIndex].Cells[4].Value.ToString(), out idCateg);
string nombreProd = tabla.Rows[e.RowIndex].Cells[1].Value.ToString();

bool nombreOK = nombreProd != "";

if (idOK && precioOK && cantOK && idcatgOK && nombreOK) mercado.modificarProducto(id, nombreProd, precioProd, cantProd, idCateg);

```

a la hora de leer los archivos si no existen por defecto se crean algunos datos de prueba, por ejemplo 1 usuario administrador y uno tipo cliente.  
para probar se puede ingresar con dni 0000 y pass admin

y 0001 cliente para probar las pantallas

```
public static List<Usuario> getAll()
{
    List<Usuario> usuarios;
    try
    {
        usuarios = JsonConvert.DeserializeObject<List<Usuario>>
(File.ReadAllText(fileName));
    }
    catch (Exception ex)
    {
        //en caso de no haber datos se genera un admin y se guarda en el archivo
        Console.WriteLine("archivo no encontrado, se inicializa un objeto vacio para productos");
        usuarios = new List<Usuario>();
        usuarios.Add(new Usuario(0, 0000, "admin", "admin", "admin@gmail.com", "admin", "admin", "000"));
        usuarios.Add(new Usuario(1, 0001, "cliente", "cliente", "cliente@gmail.com", "cliente", "cliente", "001"));
        saveAll(usuarios);
    }

    return usuarios;
}
```

En caso de no existir la carpeta por defecto se crea al iniciar el programa

```
static LocalFileManager()
{
    userpath = Environment.GetFolderPath(
Environment.SpecialFolder.ApplicationData) + @"\\Mercado";

    if (!Directory.Exists(userpath))
    {
        Directory.CreateDirectory(userpath);
    }
}
```

## Pantallas

principal

FPrincipal - [FLogin]

Ingrese usuario (n° DNI)

Ingrese contraseña

[Iniciar Sesión](#) [Regístrate](#) [Configurar](#)

## configuración

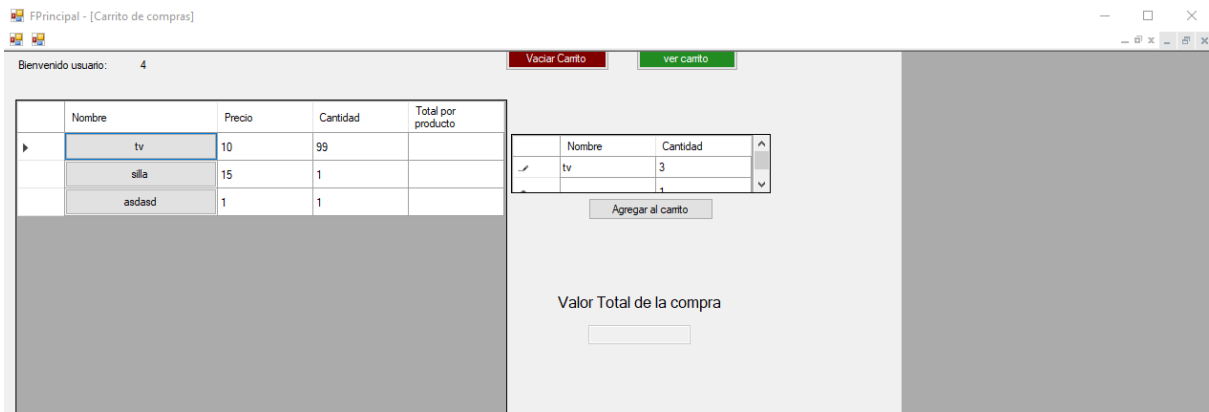
FConfigurar

Seleccione la ruta indicada para guardar los archivos

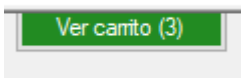
[Seleccionar](#)

[Volver atrás](#)

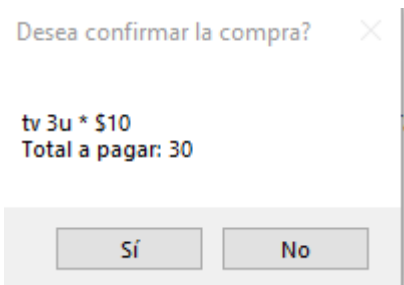
usuario



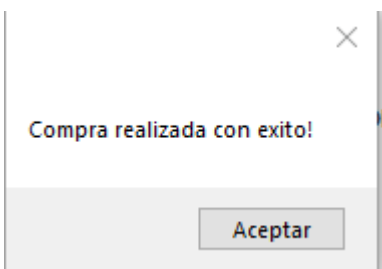
carrito mostrando cantidad de productos



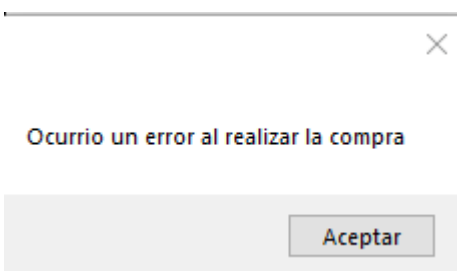
confirmacion de compra



mensaje si la compra se pudo realizar



mensaje si la compra no se puede realizar



registrar

Registro de Usuario

Nombre	<input type="text"/>	Ingrese el CUIT/CUIL	<input type="text"/>
Apellido	<input type="text"/>	Seleccione el tipo de usuario:	<input type="text" value="v"/>
DNI	<input type="text"/>	Ingrese la contraseña	<input type="password"/>
mail	<input type="text"/>		

ejemplo logueo ok

Ingrese usuario (n° DNI)

Ingrese contraseña

Log-in correcto, Usuario: admin

ejemplo logueo no ok

Ingrese usuario (n° DNI)

123

Ingrese contraseña

1231234123

Iniciar Sesion

Registrarse

Configuración

Ocurrio un error al intentar realizar el login

Aceptar

abm usuarios

FPrincipal - [FAdmin]

Bienvenido usuario: admin

Tabla\_Usuarios

Agregar

	ID	DNI	Nombre	Apellido	Mail	Password	Tipo	Cuil-Cuit	editar	borrar
▶	0	0	admin	admin	admin@gmail.com	admin	admin	000	editar	borrar
	1	1	cliente	cliente	cliente@gmail.com	cliente	cliente	001	editar	borrar
	2	0	lucas	miranda	test	123	02	cliente	editar	borrar
	3	3	3	3	3	3	3	3	editar	borrar
	4	4	4	4	44	4	4	cliente	editar	borrar
	5	1111	1111	1111	111	111	admin	111	editar	borrar
*										

abm productos

Bienvenido usuario: admin

Tabla\_Productos

Agregar

	ID	Nombre	Precio	Cantidad	Categoria	editar	borrar
▶	0	tv	10	99	0	editar	borrar
	1	silla	15	1	0	editar	borrar
	2	asdasd	1	1	1	editar	borrar
*							

abm categorias



	ID	Nombre	editar	borrar
▶	0	electro	editar	borrar
	1	deco	editar	borrar
	2	0	editar	borrar
	3	3	editar	borrar
	4	4	editar	borrar
	5	1111	editar	borrar
	6	222	editar	borrar
	7	2312312312312	editar	borrar
*				

## abm compras

Bienvenido usuario: admin Tabla Compras Agregar

	ID	info	editar	borrar
▶	7507	Usuario: 0 - Dni 0...	editar	borrar
*		Usuario: 0 - Dni 0 - Nombre admin - Apellido admin - Mail admin@gmail.com - CUIL/CUIT:000 - Tipo de Usuario:admin		