



# Proyecto DAM MolíGest

---

Autor: **Miguel Ángel Petrus Jordà**  
Tutor: **Eva Gavidia Cantavella**  
Convocatoria: **2º semestre 2023-2034**



# Contenido de la presentación

---

- **Introducción**
- **Contexto**
- **Análisis**
- **Diseño**
- **Conclusión**





# Motivación



# Motivación MolíGest

---

- Digitalizar todos los procesos de la AAVV (compras, productos, terceros, eventos, socios, asistentes a eventos)
- Tener más y mejor información acerca de la situación de la AAVV
- Mayor control y mejor explotación de los recursos de los que dispone la AAVV.
- Fomentar la participación de socios y vecinos de la AAVV

# ¿Por qué MolíGest?

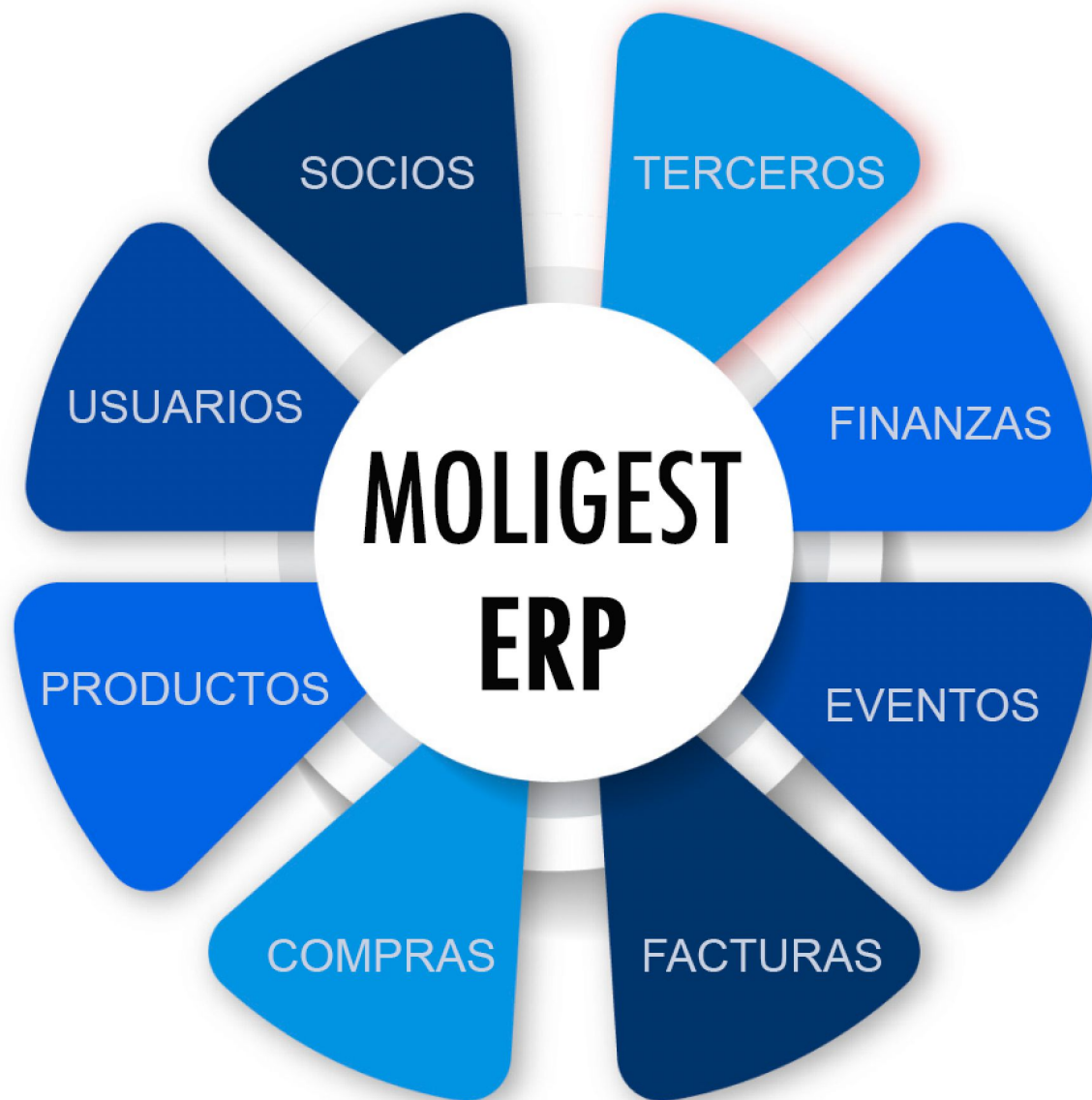
---

Opciones ERP analizadas



## Motivos MolíGest

- Mayor personalización para la AAVV
- Mayor control
- Menor coste
- Mayor versatilidad de cara a futuro



# Qué es MolíGest

---

Software de gestión integral para  
**AAVV**

# Qué podemos hacer con MolíGest

---

- Gestión de **SOCIOS** (altas, bajas)
- Gestión de **EVENTOS** (participantes, compras, ingresos..)
- Gestión de **PRODUCTOS, FAMILIAS DE PRODUCTOS**
- Gestión y control de **FINANZAS** (Ingresos, gastos, facturas..)
- Control de **BANCOS** (saldos, balances..)
- Gestión de **TERCEROS**



# Objetivos

---

**Mejorar su presentación**



# Objetivos (Generales y Específicos)

---

## Objetivos Generales

Mejorar la gestión de la AAVV "Molí des Pla" mediante:

- eficiencia
- administración de recursos
- administración de eventos y comunicaciones internas
- facilitar la toma de decisiones informada con herramientas analíticas y de informes

## Objetivos Específicos

- Implementar soluciones para gestión de eventos, recursos, finanzas
- Sistema de gestión de documentos (Facturas de compra)
- Herramienta de Análisis y Reportes
- ~~Funcionalidades de comunicación y recordatorio~~



# Tecnologías utilizadas

---

Mejorar su presentación

# Tecnologías utilizadas en MolíGest

---

## Desarrollo



## Persistencia



## IDEs



# Otras Herramientas utilizadas

---



**OBS**  
Open Broadcaster Software



# Metodología

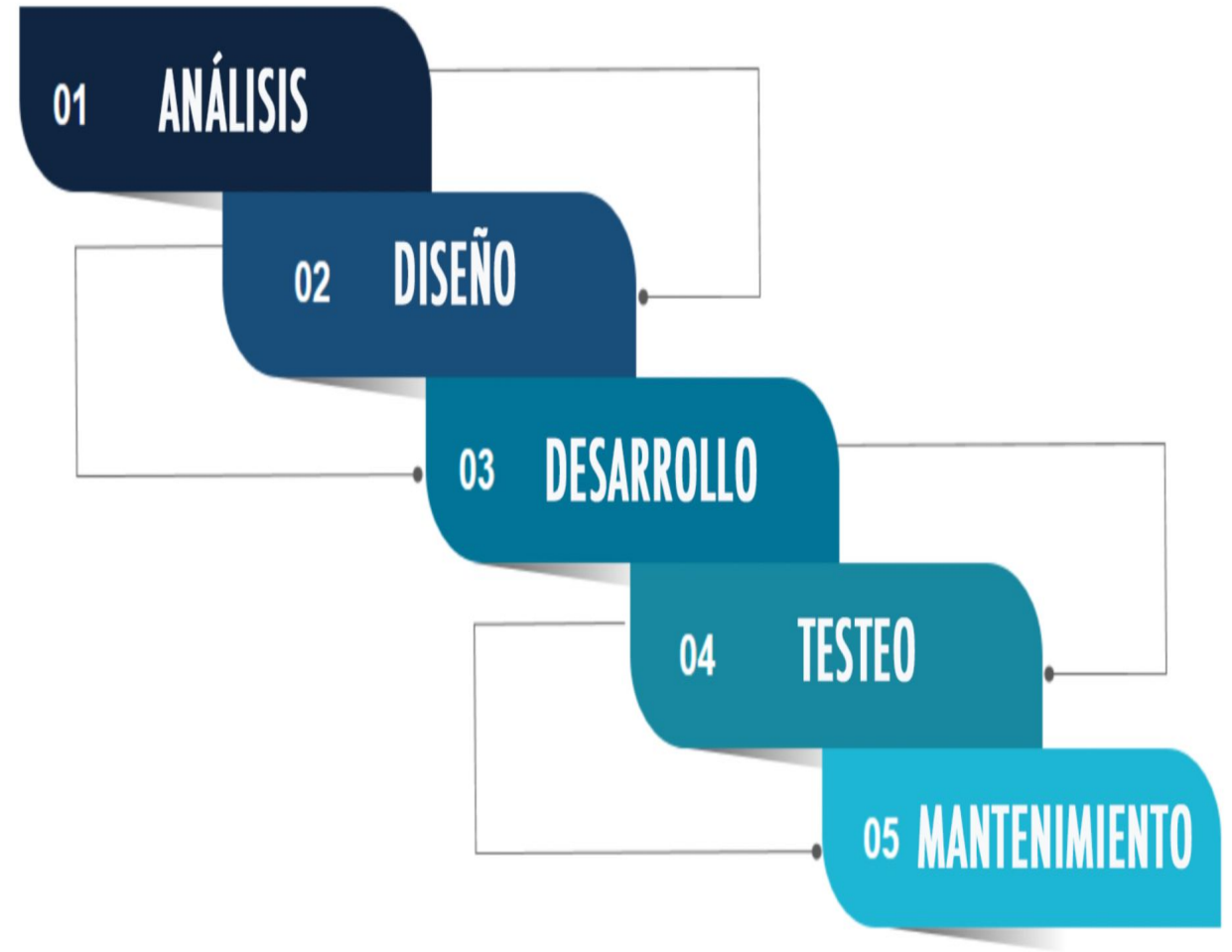
---

**Mejorar su presentación**

# Metodología en cascada

---

- Poco tiempo
- Requisitos claros



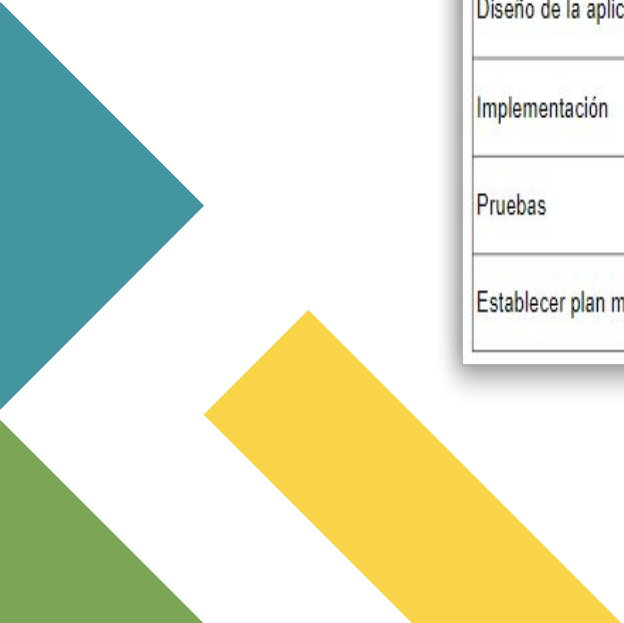


# Desarrollo proyecto

---

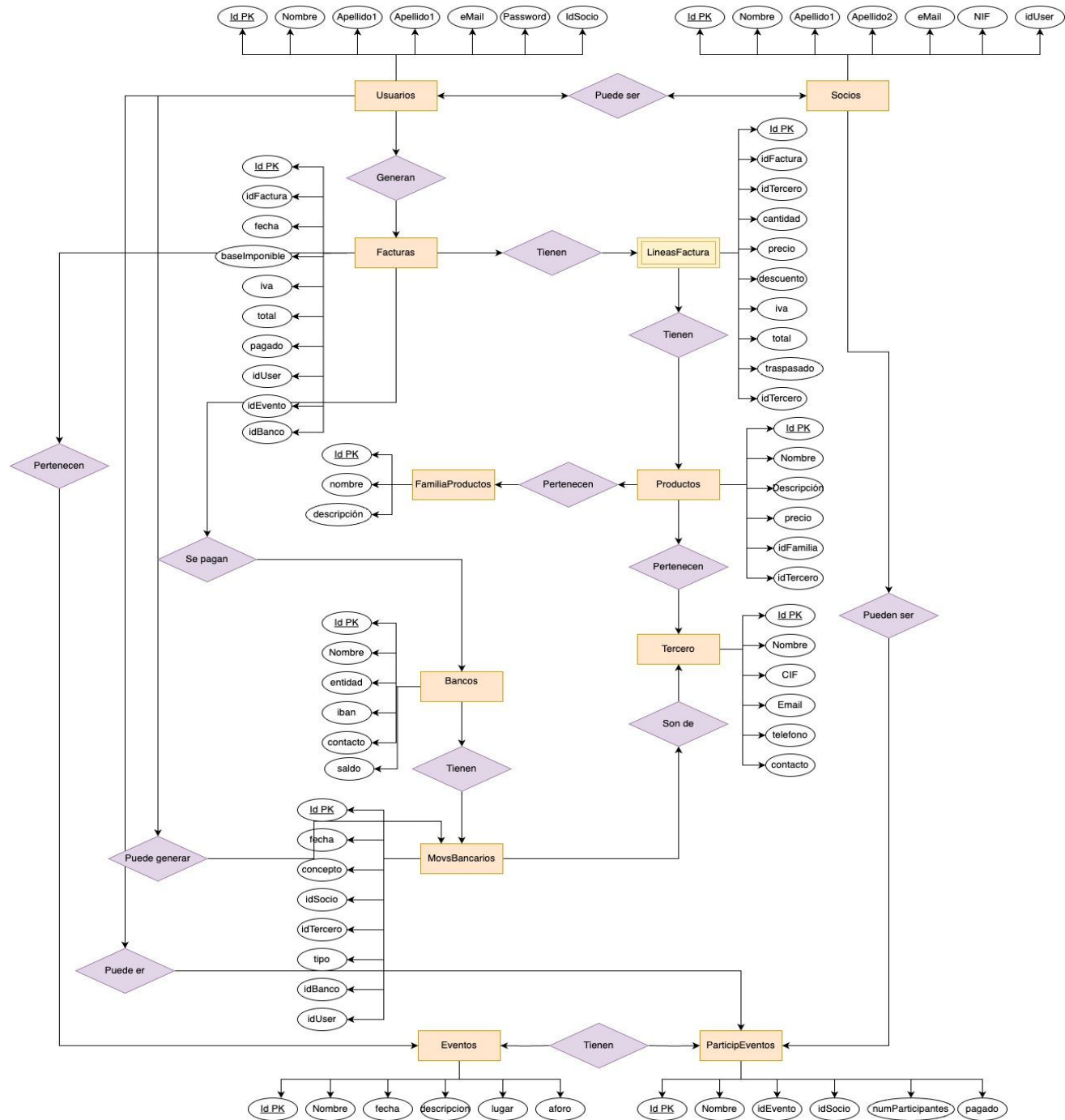
**Mejorar su presentación**

\_\_\_\_\_





# Diagrama E-R MolíGest

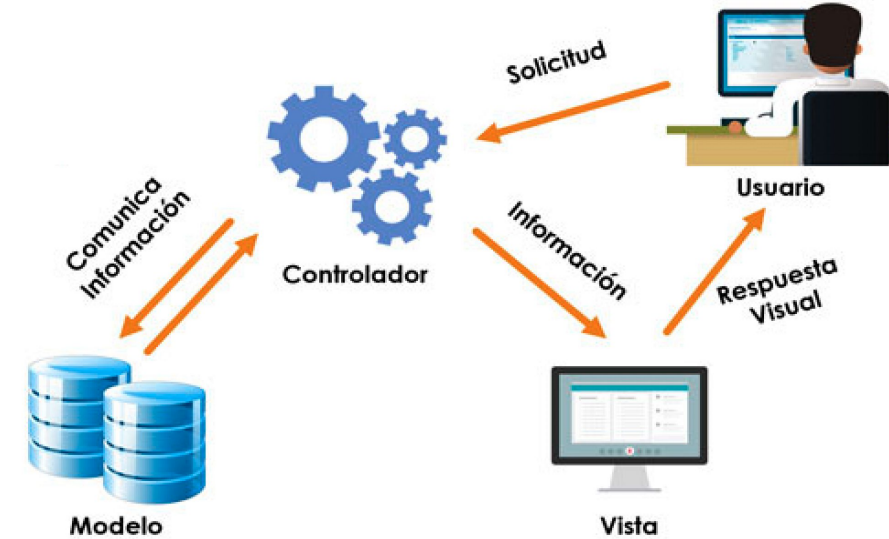


# Modelo - Vista - Controlador

## Arquitectura

### Arquitectura MVC

- lógica de negocio (Modelo)
- presentación (Vista)
- interacción del usuario (Controlador)



```
> clases
> Bancos.java
> Eventos.java
> Facturas.java
> FamiliaProducto.java
> Gastos.java
> Ingresos.java
> LineasFacturas.java
> MovimientosBancarios.java
> ParticipEventos.java
> Productos.java
> Roles.java
> Socios.java
> Terceros.java
> Users.java
```

```
> controllers
> BancosController.java
> EventosController.java
> FacturasController.java
> FamiliaProductoController.java
> HibernateUtil.java
> MovBancariosController.java
> ProductosController.java
> SociosController.java
> TercerosController.java
> UsersController.java
> UsuarioSesion.java
```

```
> views
> AddBancoView.java
> AddEventosView.java
> AddFacturaView.java
> AddFamiliaProductoView.java
> AddGastoView.java
> AddIngresoView.java
> AddProductosView.java
> AddSociosView.java
> AddTercerosView.java
> AddUsersView.java
> BancosView.java
> ConfiguracionView.java
> EditarProductosView.java
> EventosView.java
> FacturasView.java
> LoginView.java
> MainApp.java
> MovBancariosView.java
> ProductosView.java
> SociosView.java
> TercerosView.java
> UsersView.java
```

1. Patrón Singleton
2. CRUD (Create, Read, Update, Delete)
3. Encriptación de datos
4. Persistencia a través de JPA (Java Persistence API) e Hibernate
5. Maven para dependencias

```
@Entity
public class Productos {

    @ManyToOne
    @JoinColumn(name = "idFamilia")
    private FamiliaProducto familiaProducto;

    public FamiliaProducto getFamiliaProducto() {
        return familiaProducto;
    }

    public void setFamiliaProducto(FamiliaProducto familiaProducto) {
        this.familiaProducto = familiaProducto;
    }

    @ManyToOne
    @JoinColumn(name = "idTercero")
    private Terceros tercero;

    public Terceros getTercero() {
        return tercero;
    }

    public void setTercero(Terceros tercero) {
        this.tercero = tercero;
    }

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String nombre;

    public static List<MovimientosBancarios> getAllIngresos(int year) {
        try (Session session = HibernateUtil.buildSessionFactory().openSession()) {
            String hql = "FROM MovimientosBancarios WHERE tipo = 'Ingreso' AND YEAR(fecha) = :year";
            Query<MovimientosBancarios> query = session.createQuery(hql, MovimientosBancarios.class);
            query.setParameter("year", year);
            return query.list();
        } catch (Exception e) {
            e.printStackTrace();
            return Collections.emptyList();
        }
    }

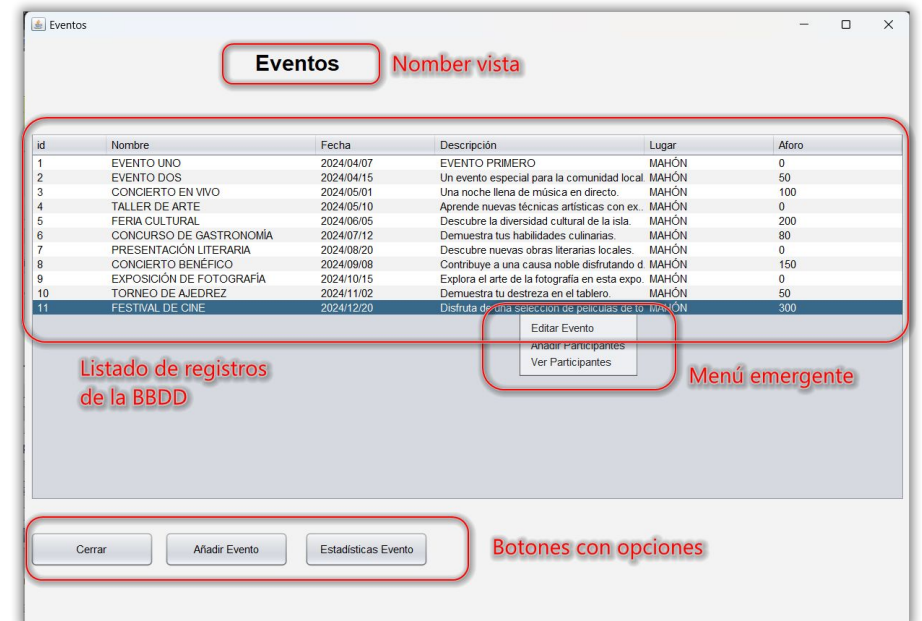
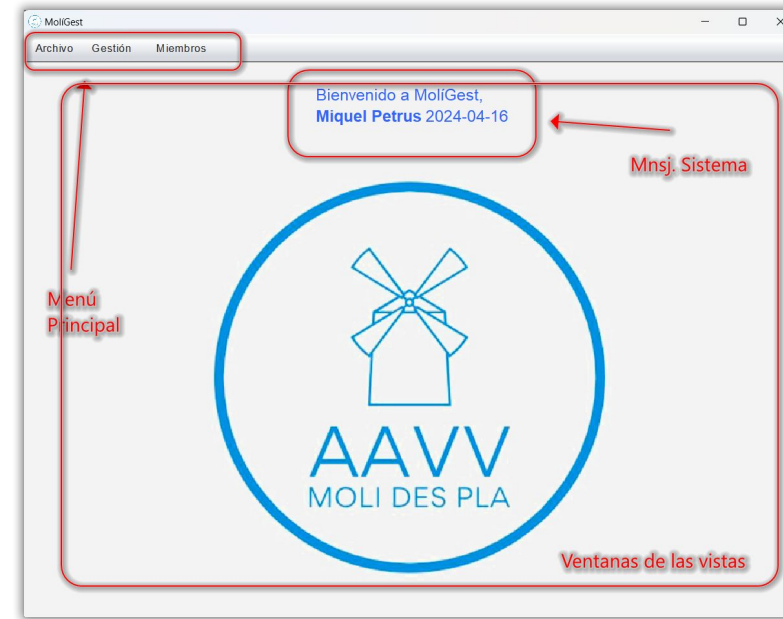
    public void run() {
        MainApp mainapp = new MainApp(); // instancia de la clase MainApp
        mainapp.setVisible(true);
        mainapp.setMenuAdminVisibility(true); // muestra las opciones de menú de administrador
        mainapp.setMenuVisibility(false); // oculta las opciones de menú hasta que el usuario se conecte
        mainapp.setLocationRelativeTo(null); // para que se muestre en el centro de la pantalla
        mainapp.setIconImage(new ImageIcon("src/main/resources/images/icono.png").getImage());
        mainapp.setTitle("MoliGest");

        LoginView loginView = new LoginView(mainapp);
        HibernateUtil.abrirVentana(loginView, "Conectar");
    }
}
```

# Soluciones utilizadas en el desarrollo

# Interfaz de Usuario

- Diseño claro y sencillo
- Disposición de menú superior
- Ventanas con título de la vista + contenido según la vista





# Conclusiones

---

**Mejorar su presentación**

# Conclusiones

Logros, Desafíos y Aprendizajes	Objetivo	Logrado
Desarrollo Efectivo (%)	100	85
Diseño centrado en el Usuario (%)	75	60
Funcionalidades clave implementadas (%)	90	85
Adaptabilidad y Flexibilidad (%)	100	95
Desafíos técnicos superados (%)	100	100

# Vías Futuras

## Interfaz Usuario

- Mejorar significativamente la UI

## Integración con página web

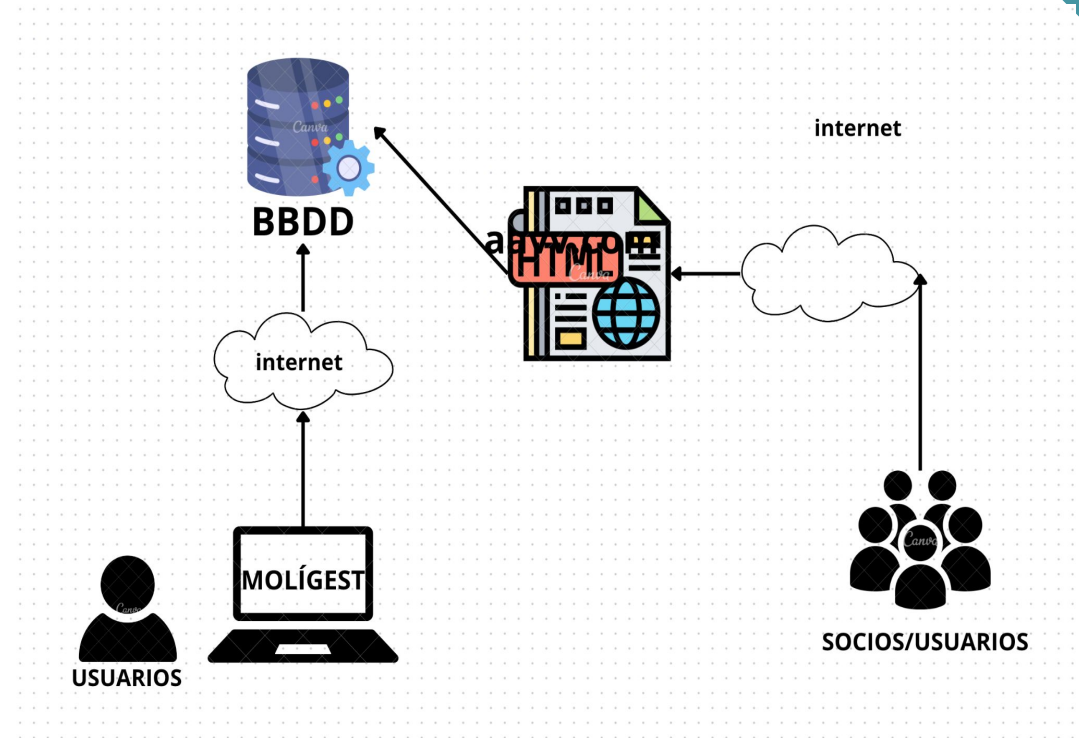
- Formularios alta socios
- Inscripciones a eventos

## CRM

- Integración de mailing para anuncios y actividades de la AAVV

## Encuestas y retroalimentación

- Encuestas a socios/no-socios





# Gracias

---

**Miguel Angel Petrus Jordà**  
**miquelpetrus@gmail.com**