

```
<!--Programacion II-->
```

```
AppBiblioteca {
```

```
<"Tomas Mastropietro"+" Juan  
Segura"+" Gianfranco Guzman">
```

```
}
```



# Introducción {

Este trabajo es sobre una app de biblioteca que hicimos en Java. Tiene una interfaz gráfica construida con JFrame, lo que hace que sea fácil de usar para cualquier persona. Con esta app se puede registrar libros y personas, hacer préstamos, extenderlos, devolver libros y ver algunos datos generales. Usamos lo que vimos en clase

- POO
- Excepciones.
- JDBC.
- GUI Swing
- Gradle.
- MySql
- IDE: IntelliJIDEA /  
Netbeans

}

# Funciones {

La función principal de nuestra app de biblioteca es gestionar de forma digital y eficiente los préstamos de libros, permitiendo a los usuarios:

- . Registrar libros y personas de forma simple.
- . Ver la disponibilidad de libros en tiempo real.
- . Realizar préstamos asociados a una persona.
  - . Extender préstamos (solo una vez por préstamo).
  - . Devolver libros y actualizarlos como disponibles.
- . Consultar préstamos activos por persona.
- . Ver todos los usuarios registrados (con acceso restringido por contraseña).



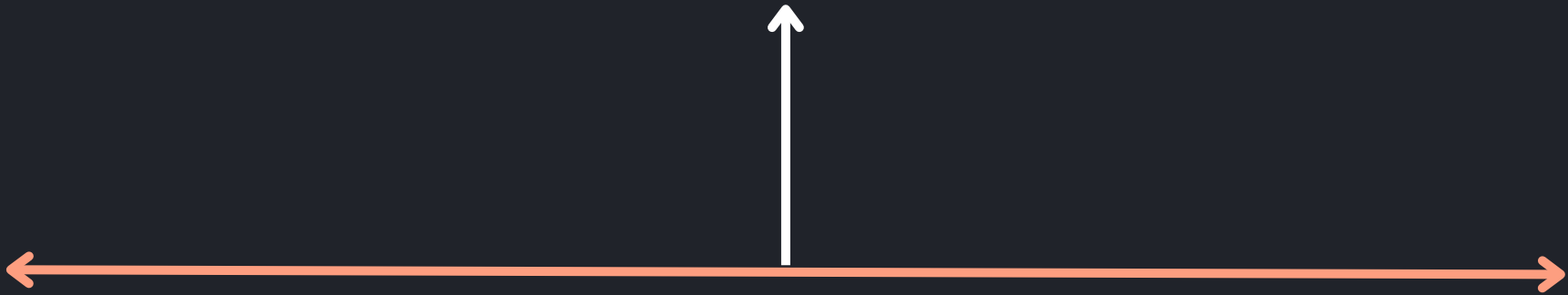
}

Estructura {

Appbiblioteca

PRESTAMO

Persona



Libro

DataBaseManger

Main

}

Puntos clave {

01

Gestión de  
biblioteca  
digital

02

Control de  
préstamos:

03

Historial  
por persona

04

Acceso  
protegido

05

Validaciones  
internas

06

Persistencia  
en base de  
datos

07

Interfaz  
amigable

08

POO

}

# El Código {

## Metodos Utilizados

- mostrarLibrosDisponibles()
- realizarPrestamo()
- extenderPrestamo()
- devolverLibro()
- verMisPrestamos()
- mostrarUsuariosRegistrados()
- 

Métodos en las clases DAO

LibroDAO

- obtenerLibrosDisponibles(Connection conn)
- insertarLibros(Connection conn)
- obtenerTituloLibro(Connection conn, int idLibro)
- 

PersonaDAO

- IngresarUsuario(Connection conn, int dni, String nombre)

PrestamoDAO

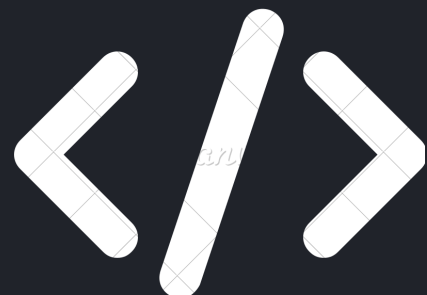
- IngresarPrestamo(Connection conn, int dni, int idLibro, String tituloLibro, Date fechaSalida, Date fechaDevolucion)
- devolverPrestamo(int idPrestamo)
- eliminarPrestamo(Connection conn, int idPrestamo)



}

## Validaciones y seguridad{

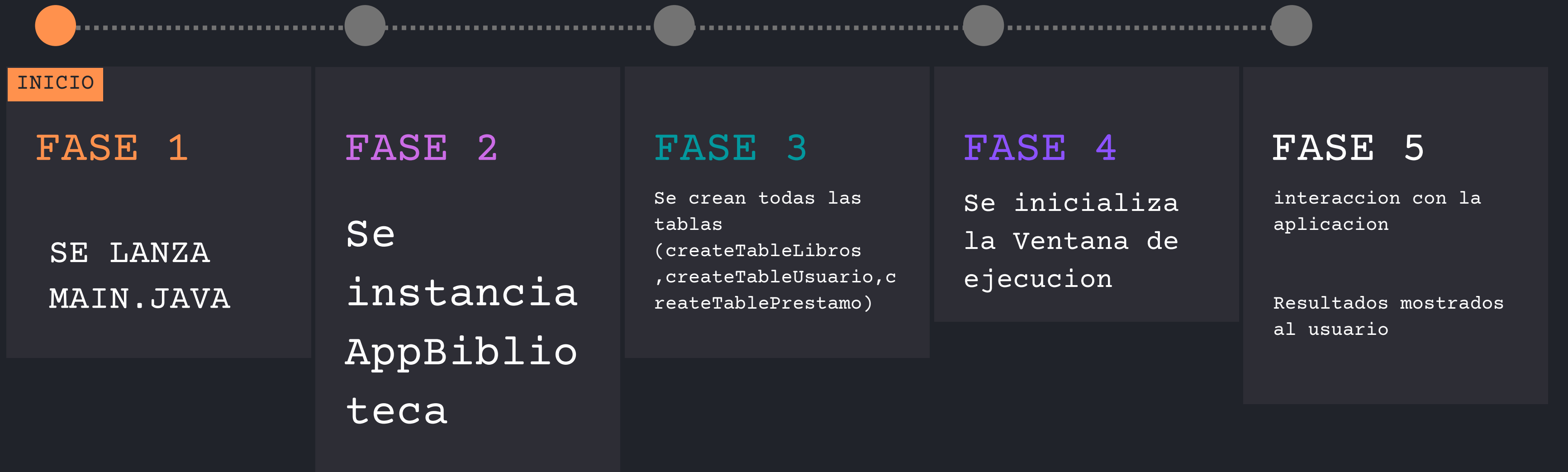
- No se pueden prestar libros que ya están prestados.
- Cada préstamo puede extenderse sólo una vez.
- Al devolver, el libro vuelve a estar disponible.



- El acceso a los usuarios está protegido con una contraseña , si se falla tres veces, la aplicación se cierra.
- Manejo de errores para entradas inválidas o IDs inexistentes.



# Flujo de sistema







# Gracias {

```
Public static NoTeDuermasTomi{  
    Int Dormir = 100  
    //no se duerman muchachos dale  
}
```

# }

