

EQUIPO

Sudo★ku

- ▶ Iteración 2
- ▶ Diseño detallado
- ▶ Versión 1.0
- ▶ 29 de Noviembre del 2022

1. Galván Solís Sabrina, responsable
2. Beristain Hernández Daniel, técnico
3. Ramirez Gutierrez Oscar, colaborador
4. Cristóbal Morales Karen, calidad
5. Martinez Enriquez Bruno, calidad
6. Ortega Garcia Alejandra, calidad



Diseño detallado de la iteración

► Contenido

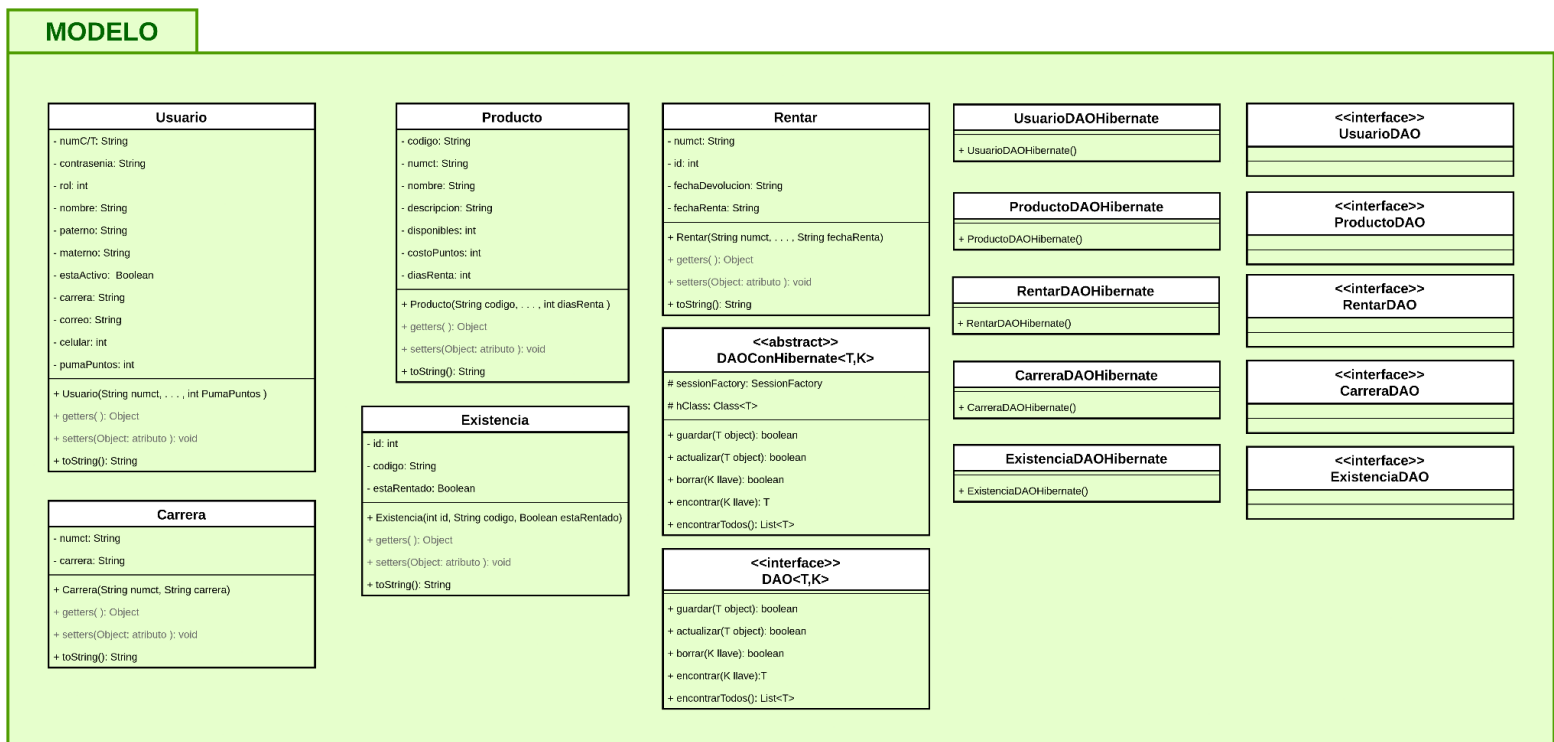
- Referencia al documento de *Especificación de requerimientos de software*
- Diagramas de clases integrados
- Diagramas de secuencia
 1. Iniciar sesión.
 2. Ver perfil.
 3. Ver reportes.

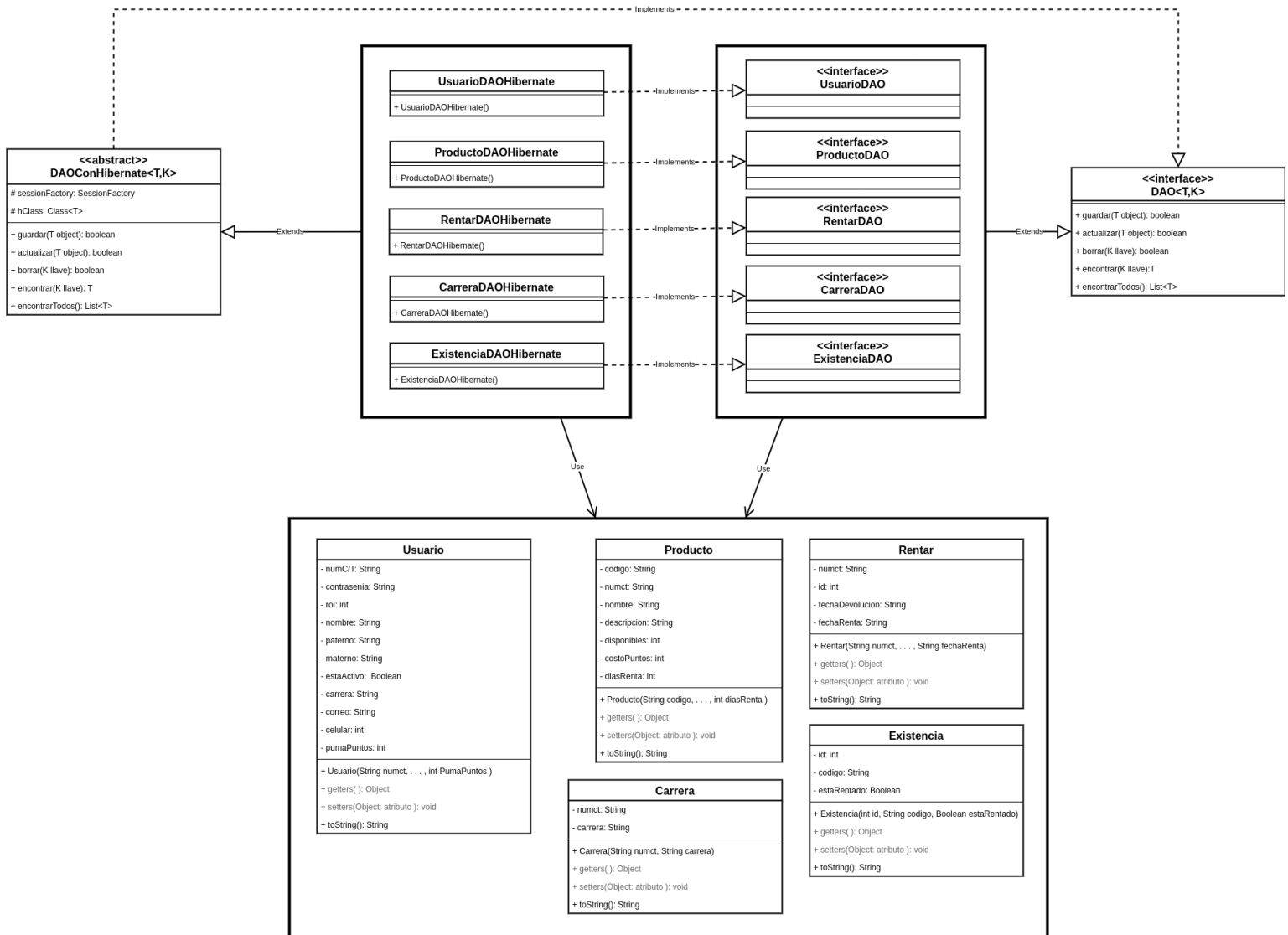


Referencia al documento de Especificación de requerimientos de software

<https://docs.google.com/document/d/1whRSBNkI5ZAlfWvq4Mw-9oNhus1cbe3N/edit?usp=sharing&oid=105447071761988781768&rtpof=true&sd=true>

Diagramas de clases integrados





Referencia: http://www.cursohibernate.es/doku.php?id=unidades:07_arquitectura:03_dao



CONTROLADOR

UsuarioControlador

```
- usuarioDAO : UsuarioDAO

+ getters( ): Object
+ setters(Object: atributo ): void
+ iniciarSesion(String numct, String contrasenia) : boolean
+ cerrarSesion(): void
+ restablecerContrasenia(Usuario usuario, String nuevaContrasenia): void
+ verPerfil(Usuario usuario): void
+ acumularPuntos(Usuario usuario, int puntos): void
+ verHistorial(Usuario usuario): List<Rentar>
+ buscarUsuario(String numct): Usuario
+ verUsuarios(): List <Usuario>
+ agregarUsuario(Usuario usuario): boolean
+ editarUsuario(Usuario usuario): boolean
+ eliminarUsuario(Usuario usuario): boolean
+ verReportes(Usuario usuario): boolean
+ sumarPuntos(Usuario usuario, int puntos): boolean
+ restarPuntos(Usuario usuario, int puntos): boolean
```

ProductoControlador

```
- productoDAO: ProductoDAO

+ getters( ): Object
+ setters(Object: atributo ): void
+ buscarProducto(String codigo): producto: Producto
+ verProductos(Usuario usuario): List<Producto>
+ verProductos(): List<Producto>
+ agregarProductos(Producto producto): boolean
+ editarProducto(Producto producto): boolean
+ eliminarProducto(Producto producto): boolean
```

RentarControlador

```
-rentarDAO : RentarDAO

+ getters( ): Object
+ setters(Object: atributo ): void
+ rentarProducto(usuario: Usuario, existencia: Existencia): boolean
+ devolverProducto(usuario: Usuario, existencia: Existencia): boolean
```

ExistenciaControlador

```
- existenciaDAO : ExistenciaDAO

+ getters( ): Object
+ setters(Object: atributo ): void
+ tenerExistencia(producto: Producto): boolean
+ agregarExistencia(Producto, producto, int existencias): boolean
+ eliminarExistencia(Producto, producto, int existencias): boolean
```

CarreraControlador

```
- carreraDAO : CarreraDAO

+ getters( ): Object
+ setters(Object: atributo ): void
+ agregarCarrera(usuario: Usuario, Carrera carrera): boolean
+ eliminarCarrera(usuario: Usuario, Carrera carrera): boolean
+ tenerCarrera(usuario: Usuario, Carrera carrera): boolean
```



VISTA

| IniciarSesionUI |
|---------------------------------|
| - usuario: Usuario |
| + iniciarSesion(): void |
| + restablecerContraseña(): void |

| MensajeUI |
|--------------------------|
| - mensaje: String |
| + mostrarMensaje(): void |

| RestablecerContraUI |
|------------------------|
| - usuario: String |
| - nuevaContra : String |
| + restablecer(): void |

Usuario Normal

| PrincipalNormalUI |
|--------------------------|
| + verPerfil(): void |
| + cerrarSesion(): void |
| + verHistorial(): void |
| + verProducto(): void |
| + buscarProducto(): void |
| + acumularPuntos(): void |

| VerProductosNormalUI |
|---------------------------|
| + rentarProductos(): void |

Proveedor

| PrincipalProveUI |
|--------------------------|
| + verPerfil(): void |
| + cerrarSesion(): void |
| + verHistorial(): void |
| + verProducto(): void |
| + buscarProducto(): void |
| + acumularPuntos(): void |

| VerProductosProveUI |
|---------------------------|
| + rentarProductos(): void |
| + agregarProducto(): void |

| BuscarProductoProveUI |
|----------------------------|
| + editarProducto(): void |
| + eliminarProducto(): void |

Administrador

| PrincipalAdmiUI |
|--------------------------|
| + verPerfil(): void |
| + cerrarSesion(): void |
| + verHistorial(): void |
| + verProducto(): void |
| + buscarProducto(): void |
| + buscarUsuario(): void |
| + verUsuarios(): void |
| + verReportes(): void |
| + acumularPuntos(): void |

| VerProductosAdmiUI |
|---------------------------|
| + rentarProductos(): void |
| + agregarProducto(): void |

| BuscarProductoAdmiUI |
|----------------------------|
| + editarProducto(): void |
| + eliminarProducto(): void |

| VerUsuariosAdmiUI |
|--------------------------|
| + agregarUsuario(): void |

| BuscarUsuarioAdmiUI |
|---------------------------|
| + editarUsuario(): void |
| + eliminarUsuario(): void |

| EditarUsuarioAdmiUI |
|------------------------|
| + sumarPuntos(): void |
| + restarPuntos(): void |

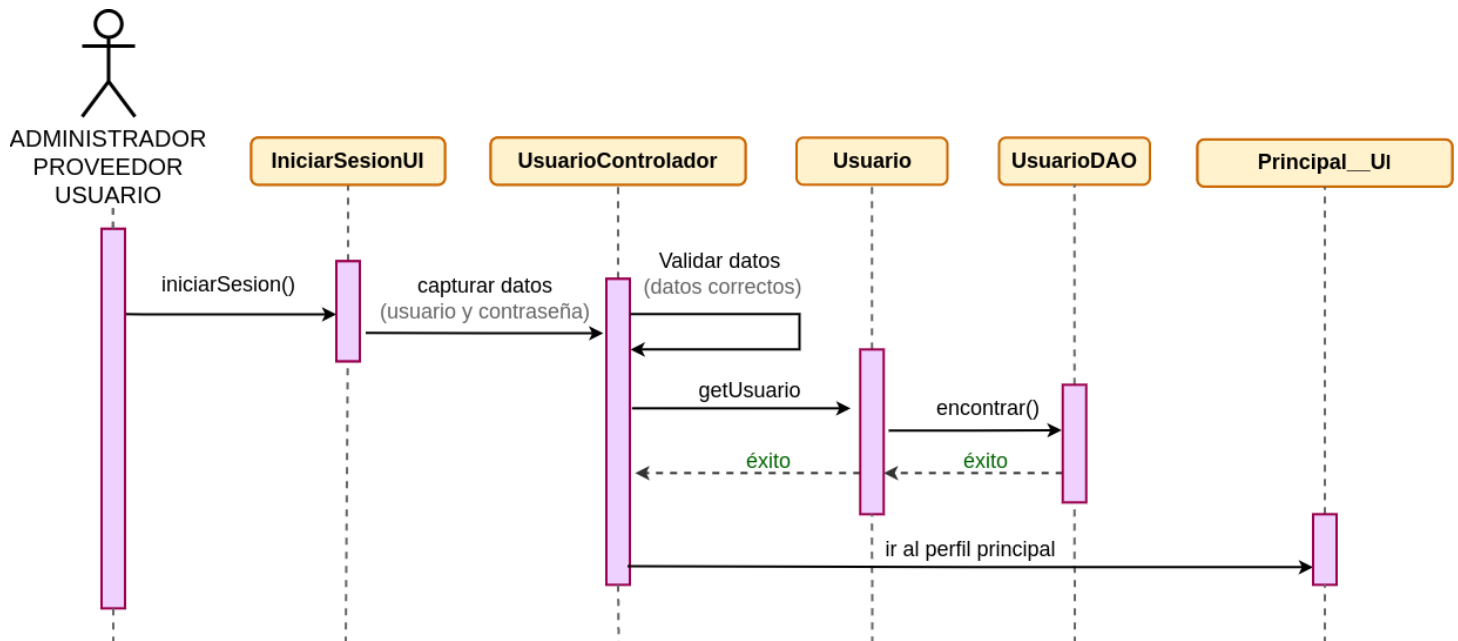
UI = User Interface



Diagramas de secuencia

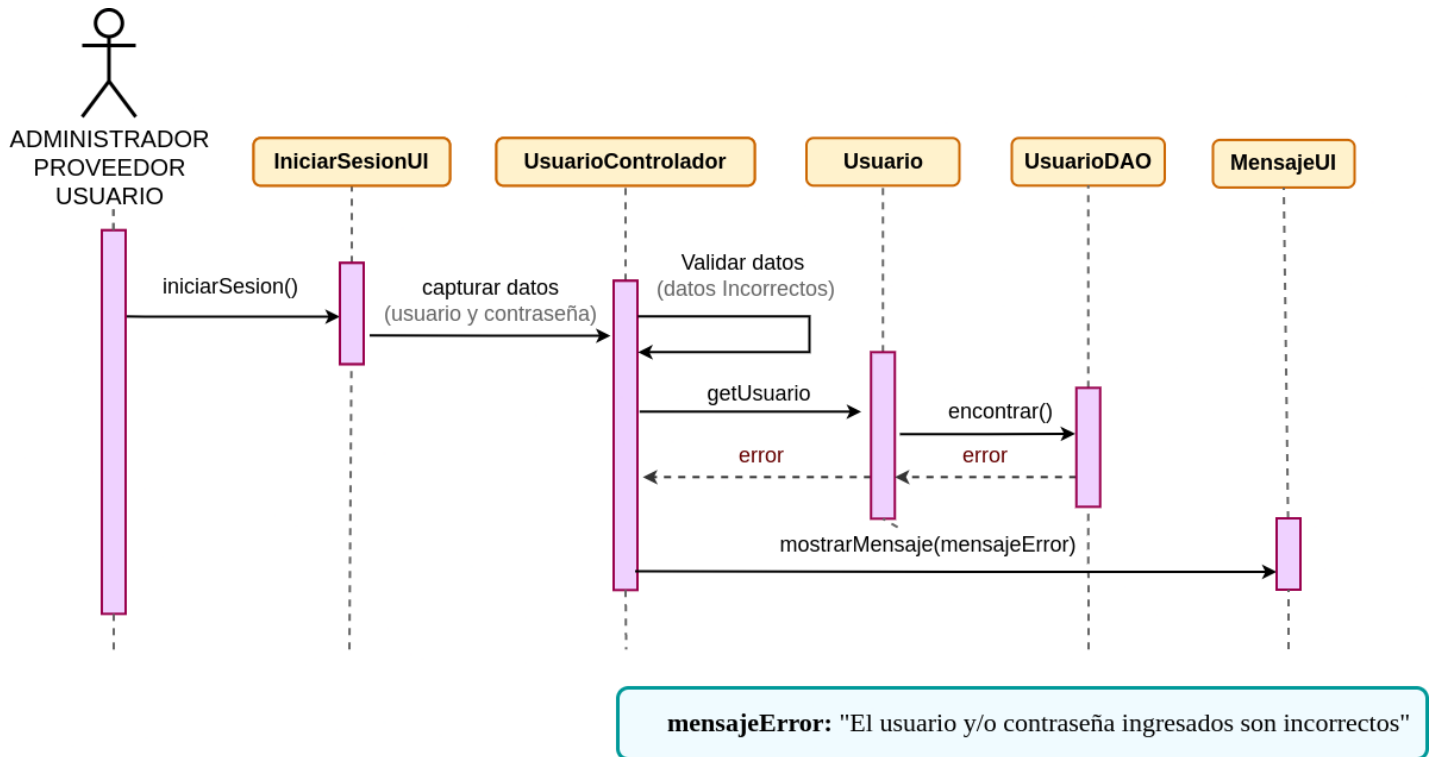
• INICIAR SESIÓN

FLUJO NORMAL DE EVENTOS:





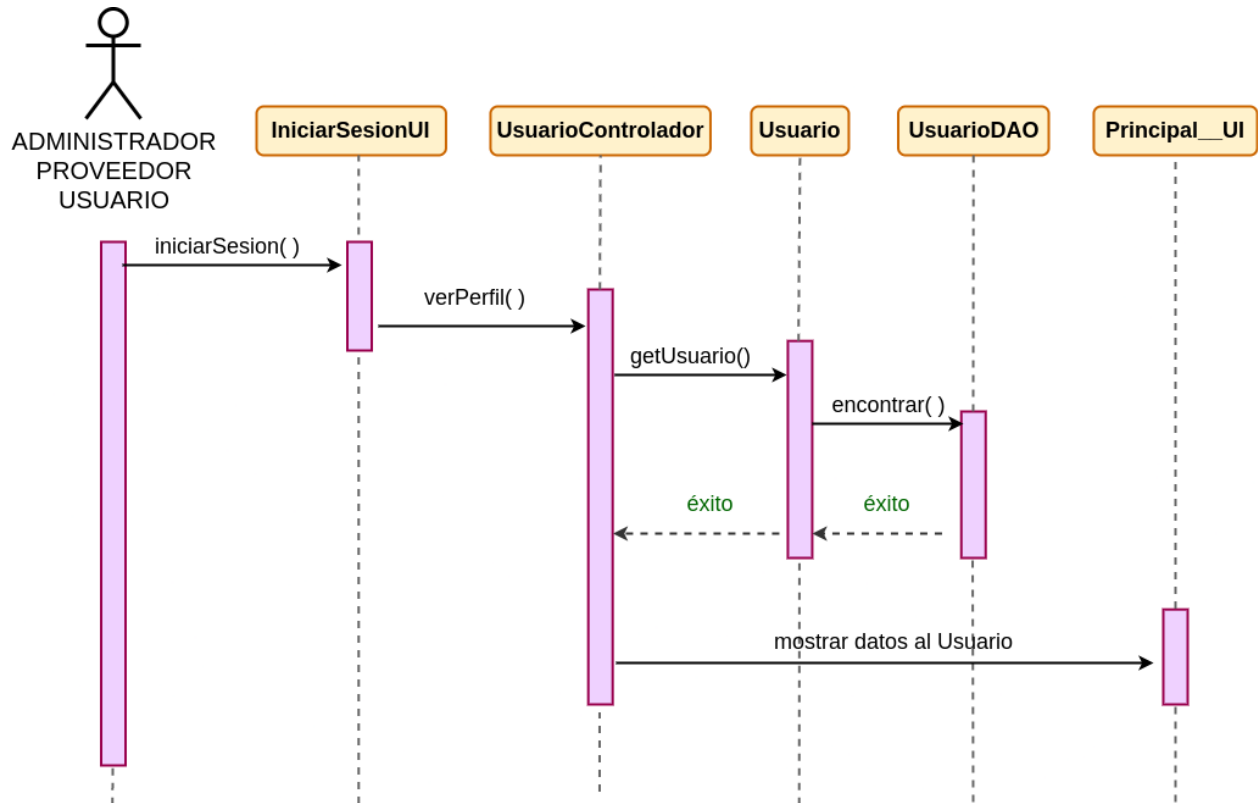
FLUJO ALTERNATIVO DE EVENTOS:





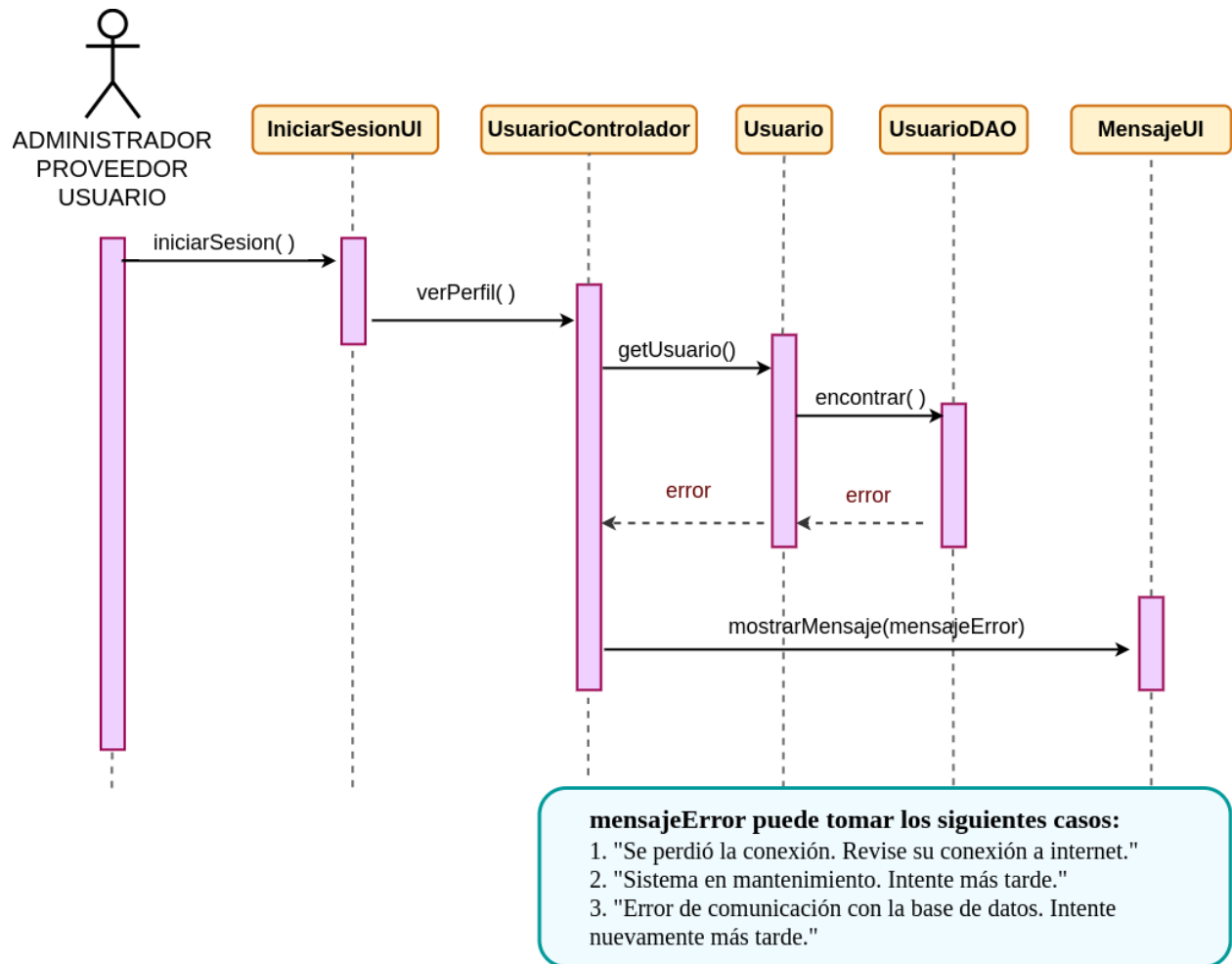
• VER PERFIL

FLUJO NORMAL DE EVENTOS:





FLUJO EXCEPCIONAL DE EVENTOS:





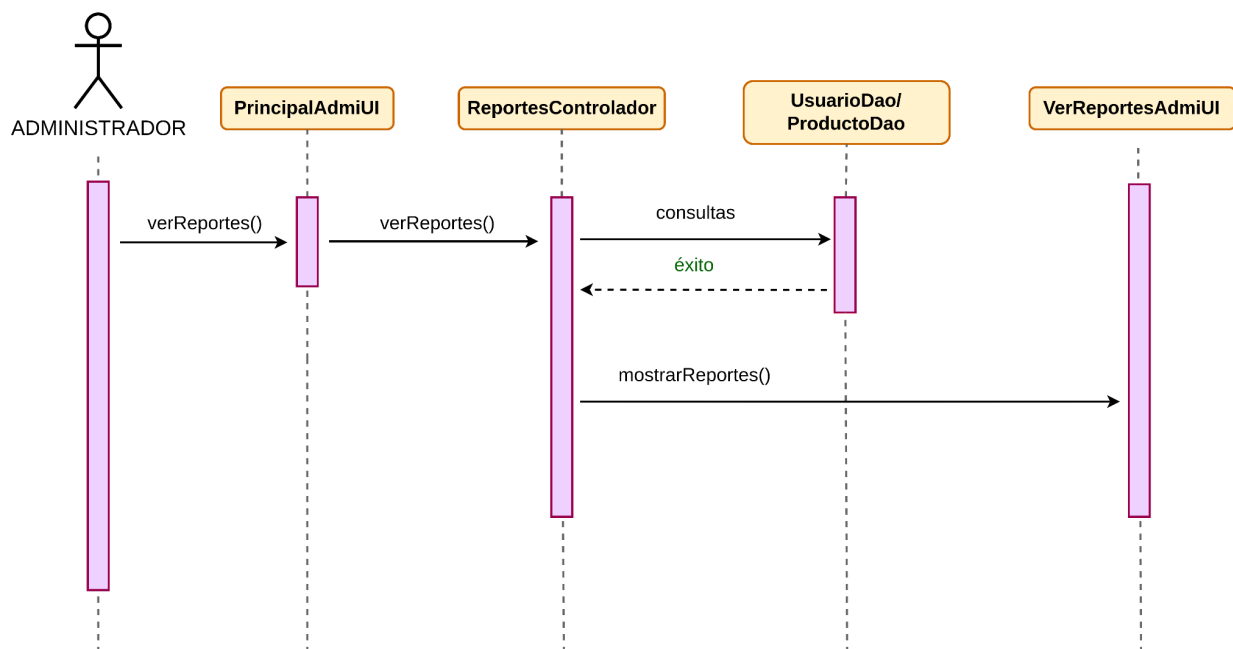
FLUJO ALTERNATIVO DE EVENTOS:

No hay ya que no se recibe alguna entrada por parte del usuario.

El único caso que podría ser “alternativo” sería que la base de datos de los productos esté vacía, es decir, aún no se ha agregado ningún producto al sistema.

• VER REPORTE

FLUJO NORMAL DE EVENTOS:



Equipo: Sudo★ku
Sistema: Ciencias Top
Iteración: 2



FLUJO ALTERNATIVO DE EVENTOS:

No hay, ya que no se recibe alguna entrada por parte del administrador.

El único caso que podría ser “alternativo” sería que la base de datos de los productos esté vacía, es decir, aún no se ha agregado ningún producto o usuario al sistema para poder generar el reporte.