***UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO***

***FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES***



***REGEVENT: CONTROL DE PASES QR***

***PATRON DE ARQUITECTURA: 3 CAPAS***

***PATRONES DE DISEÑO: OBSERVER Y STRATEGY***

**Estudiante:** Herbas Miranda José Mario

**Nro Registro:** 220001448

**Docente:** Ing. Josué Veizaga

**Materia:** Arquitectura de Software

**Semestre:** 1-2024

**Fecha de entrega:** 14/06/2024

***SANTA CRUZ DE LA SIERRA, BOLIVIA***

Tabla de contenido

[CONTENIDO 4](#_Toc169232468)

[1. Flujo: Requisitos 4](#_Toc169232469)

[1.1. Requisitos funcionales 4](#_Toc169232470)

[1.2. Requisitos no funcionales 5](#_Toc169232471)

[1.3. Identificar casos de uso 6](#_Toc169232472)

[Caso de Uso 1.- Gestionar Eventos 6](#_Toc169232473)

[Caso de Uso 2.- Gestionar Invitación 8](#_Toc169232474)

[Caso de Uso 3.- Gestionar Asistencia 10](#_Toc169232475)

[Caso de Uso 4.- Gestionar Mesa 12](#_Toc169232476)

[Caso de Uso 5.- Gestionar Silla 14](#_Toc169232477)

[2. FLUJO: ANALISIS 16](#_Toc169232478)

[2.1. IDENTIFICACION DE LOS MODULOS 16](#_Toc169232479)

[2.1.1. Módulo de Participantes 16](#_Toc169232480)

[2.1.2. Módulo de Planificación 16](#_Toc169232481)

[2.2. VISTA DE LOS MODULOS 16](#_Toc169232482)

[2.2.1. Vista del módulo de Participantes 16](#_Toc169232483)

[2.2.2. Vista del módulo de Planificación 17](#_Toc169232484)

[3. FLUJO: DISEÑO 18](#_Toc169232485)

[3.1. DISEÑO DE LA ARQUITECTURA DE SOFTWARE 18](#_Toc169232486)

[3.1.1. Módulo de Participantes 18](#_Toc169232487)

[3.1.2. Módulo de Planificación 18](#_Toc169232488)

[3.2. DISEÑO DE LA BASE DE DATOS 18](#_Toc169232489)

[4.2.1 Diseño Conceptual de la Base de Datos 18](#_Toc169232490)

[3.2.2. Diseño Lógico de la Base de Datos 19](#_Toc169232491)

[3.2.3. Diseño Físico de la Base de Datos 19](#_Toc169232492)

[3.3. DISEÑO DE LA INTERFAZ 22](#_Toc169232493)

[3.4. DISEÑO DE DETALLE PROCEDIMENTAL 22](#_Toc169232494)

[3.5. DIAGRAMA DE SECUENCIA 22](#_Toc169232495)

[CU1. Gestionar Evento 22](#_Toc169232496)

[CU2. Gestionar Invitación 26](#_Toc169232497)

[CU3. Gestionar Asistencia 32](#_Toc169232498)

[CU4 Gestionar Mesa 35](#_Toc169232499)

[CU5 Gestionar Silla 38](#_Toc169232500)

[4. CODIGO 41](#_Toc169232501)

[4.1. CU1. GESTIONAR EVENTO 41](#_Toc169232502)

[4.2. CU2. GESTIONAR INVITACION 41](#_Toc169232503)

[4.3. CU3. GESTIONAR ASISTENCIA 41](#_Toc169232504)

[4.4. CU4. GESTIONAR MESA 42](#_Toc169232505)

[4.5. CU5. GESTIONAR SILLA 42](#_Toc169232506)

[5. BIBLIOGRAFIA 42](#_Toc169232507)

# CONTENIDO

## Flujo: Requisitos

### Requisitos funcionales

1. Registro de Eventos:

* El usuario puede registrar eventos proporcionando detalles como título, ubicación, descripción, fecha y hora.
* El usuario puede cambiar los datos del evento según la necesidad.
* El usuario puede eliminar el evento, borrando así todos los componentes asociados a él como ser sillas, mesas.

1. Gestión de Invitaciones:

* El usuario puede crear invitaciones para un evento específico, incluyendo el nombre del invitado y su número de celular.
* Las invitaciones pueden ser editadas para cambiar los detalles del invitado o la mesa asignada.
* Las invitaciones pueden ser eliminadas si ya no son necesarias.

1. Gestión de Mesas:

* El usuario puede crear mesas para un evento, especificando el tipo y la capacidad de la mesa.
* Las mesas pueden ser editadas para cambiar su capacidad o tipo.
* Las mesas pueden ser eliminadas si ya no son necesarias.

1. Registro de Asistencia:

* El usuario puede registrar la asistencia de los invitados en un evento utilizando códigos QR generados para cada invitación.

1. Compartir Invitaciones:

* El usuario puede compartir las invitaciones con los invitados a través de mensajes de texto o redes sociales.

### Requisitos no funcionales

1. Seguridad:

* Los datos de los usuarios, eventos e invitaciones deben ser almacenados de manera segura para proteger la privacidad y evitar accesos no autorizados.

1. Rendimiento:

* El sistema debe ser capaz de manejar un gran número de invitaciones y asistentes simultáneos sin experimentar una degradación significativa del rendimiento.

1. Usabilidad:

* La interfaz de usuario debe ser intuitiva y fácil de usar para que los usuarios puedan navegar y utilizar las funciones del sistema sin dificultad.

1. Disponibilidad:

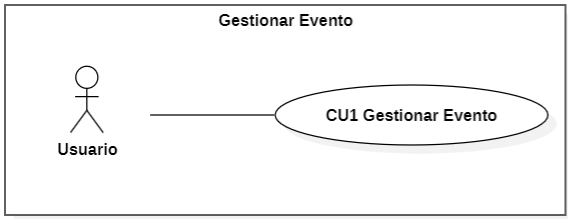
* El sistema debe estar disponible y accesible en todo momento, evitando periodos prolongados de inactividad que puedan afectar la experiencia del usuario.

1. Escalabilidad:

* El sistema debe ser escalable para poder manejar un aumento en el número de eventos, invitaciones y usuarios sin necesidad de cambios significativos en la arquitectura o la infraestructura subyacente.

### Identificar casos de uso

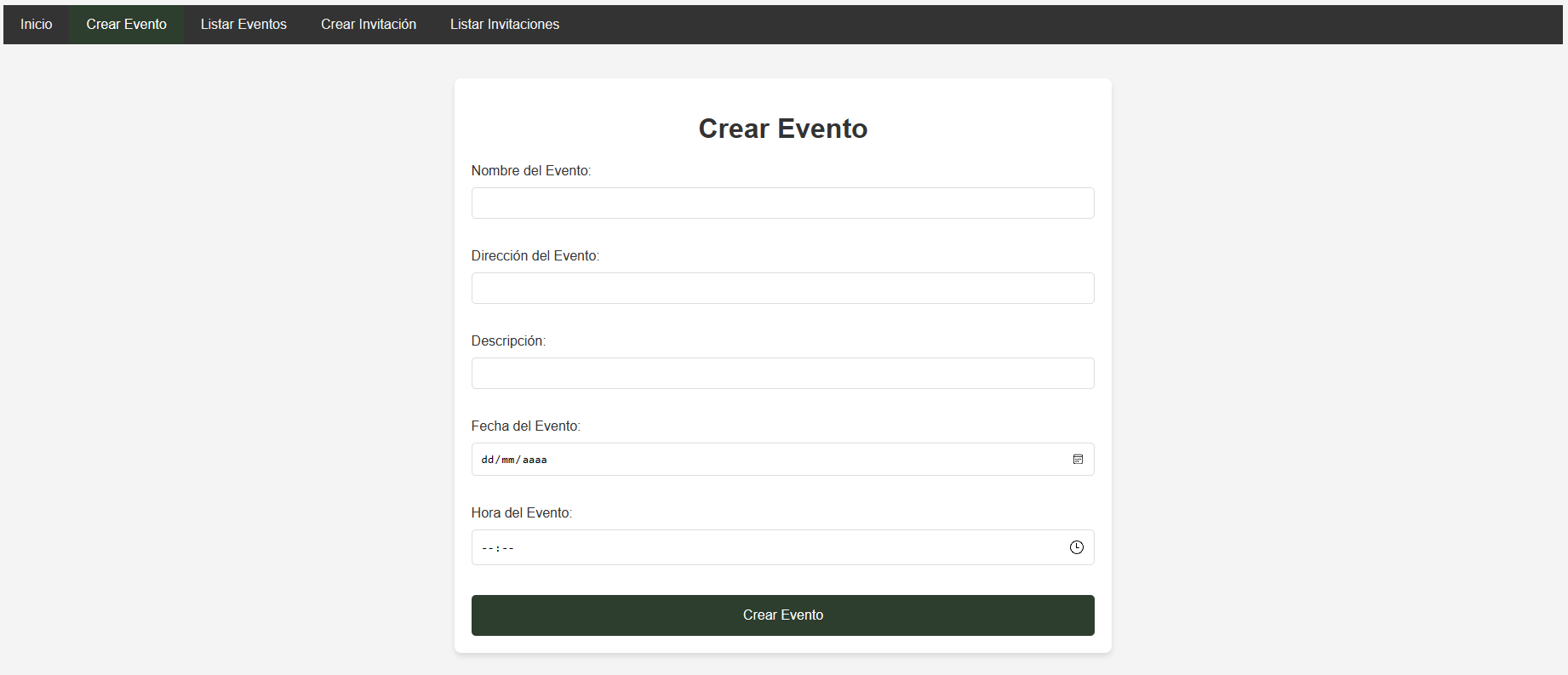
#### Caso de Uso 1.- Gestionar Eventos



##### Detallar Caso de Uso

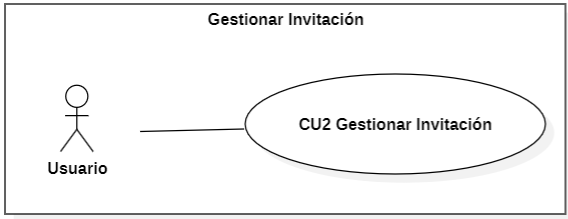
|  |  |
| --- | --- |
| **Nombre del caso de uso** | CU1. Gestionar Evento |
| **Propósito** | Permite gestionar los eventos que requiera el usuario administrador |
| **Actores** | Usuario |
| **Flujo Principal** | **Respuesta del Sistema** |
| 1. **Crear Evento** 2. Click en “Crear Evento” en el menú de navegación. 3. Rellenar los datos para crear el evento. 4. Pulsar el botón “Crear Evento”. | El sistema crea el nuevo evento y envía un mensaje: “El evento se creó correctamente.” |
| 1. **Listar Eventos.** 2. Click en “Listar Eventos” en el menú de navegación. | El sistema obtiene y muestra todos los eventos creados por el administrador. |
| 1. **Editar Eventos.** 2. Click en “Listar Eventos” en el menú de navegación. 3. Click en “Editar” del evento deseado. 4. Rellenar y actualizar los datos del Evento elegido. 5. Pulsar el botón “Guardar Cambios”. | El sistema actualiza los datos del evento elegido para su edición. |
| 1. **Eliminar Evento.** 2. Click en “Listar Eventos” en el menú de navegación. 3. Click en “Eliminar” del evento deseado. 4. Click en “Eliminar”. | El sistema eliminará el evento y todas sus invitaciones y asistencias que pertenecían al evento. |
| **Excepciones** | No rellenar algún campo de los datos. |

##### Prototipo de la Interfaz





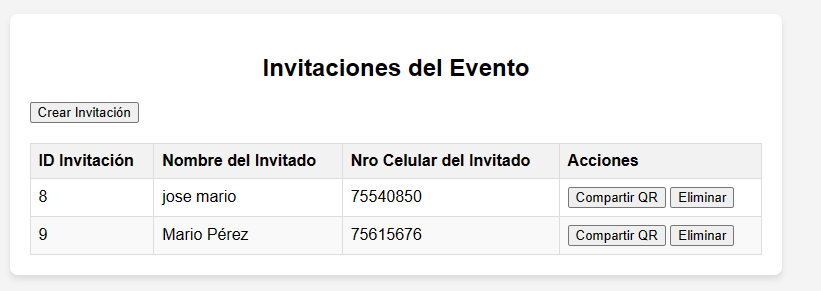
#### Caso de Uso 2.- Gestionar Invitación



##### Detallar Caso de Uso

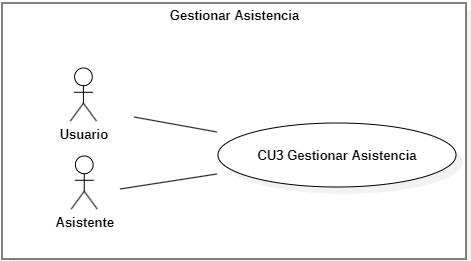
|  |  |
| --- | --- |
| **Nombre del caso de uso** | CU2. Gestionar Invitación |
| **Propósito** | Permite gestionar las invitaciones de eventos que requiera el usuario administrador. |
| **Actores** | Usuario |
| **Flujo Principal** | **Respuesta del Sistema** |
| 1. **Listar Invitaciones** 2. Click en “Ver Invitaciones” del evento deseado. | El sistema obtiene y muestra todas las invitaciones creadas por el administrador de dicho evento. |
| 1. **Crear Invitación.** 2. Click en “Crear Invitación” del evento deseado. 3. Rellenar todos los datos de la invitación. 4. Pulsar el botón “Crear Invitación”. | El sistema crea la invitación con todos los datos proporcionados. |
| 1. **Compartir Invitación por Whatsapp.** 2. Click en “Compartir QR” de la invitación deseada. 3. Se abre la aplicación Whatsapp y se envía el mensaje. | El sistema generará un mensaje estructurado y abrirá Whatsapp para que el usuario pueda enviar el mensaje al destinatario deseado. |
| 1. **Editar Invitación** 2. Click en “Editar” de la invitación deseada. 3. Actualizar los datos de la invitación. 4. Click en “Guardar cambios” | El sistema actualizará los datos de la invitación con los nuevos datos proporcionados. |
| 1. **Eliminar Invitación** 2. Click en “Eliminar” de la invitación deseada. 3. Confirmar la eliminación de la invitación pulsando el botón “Eliminar”. | El sistema eliminará la invitación y también eliminará su asistencia del usuario en caso de que haya asistido al evento. |
| **Excepciones** | No rellenar algún campo de los datos. |

##### Prototipo de la Interfaz





#### Caso de Uso 3.- Gestionar Asistencia



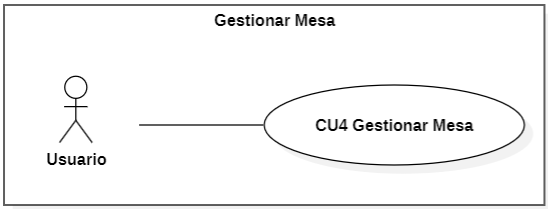
##### Detallar Caso de Uso

|  |  |
| --- | --- |
| **Nombre del caso de uso** | CU3. Gestionar Asistencia |
| **Propósito** | Permite gestionar los asistentes invitados de los eventos creados por el administrador. |
| **Actores** | Usuario, Asistente |
| **Flujo Principal** | **Respuesta del Sistema** |
| 1. **Listar Asistentes** 2. Click en “Registrar Asistencia” del evento deseado. | El sistema obtiene y muestra todos los asistentes registrados por el administrador de dicho evento. |
| 1. **Registrar Asistencia.** 2. Click en “Registrar Asistencia” del evento deseado. 3. Se abre la cámara. 4. El asistente muestra el código QR a la cámara. 5. Se registra la asistencia automáticamente. 6. Se actualiza la tabla de asistentes registrados automáticamente. | El sistema crea la asistencia del invitado.  El sistema actualiza la tabla de asistentes en tiempo real. |
| **Excepciones** | Mostrar un QR inválido. |

##### Prototipo de la Interfaz



#### Caso de Uso 4.- Gestionar Mesa

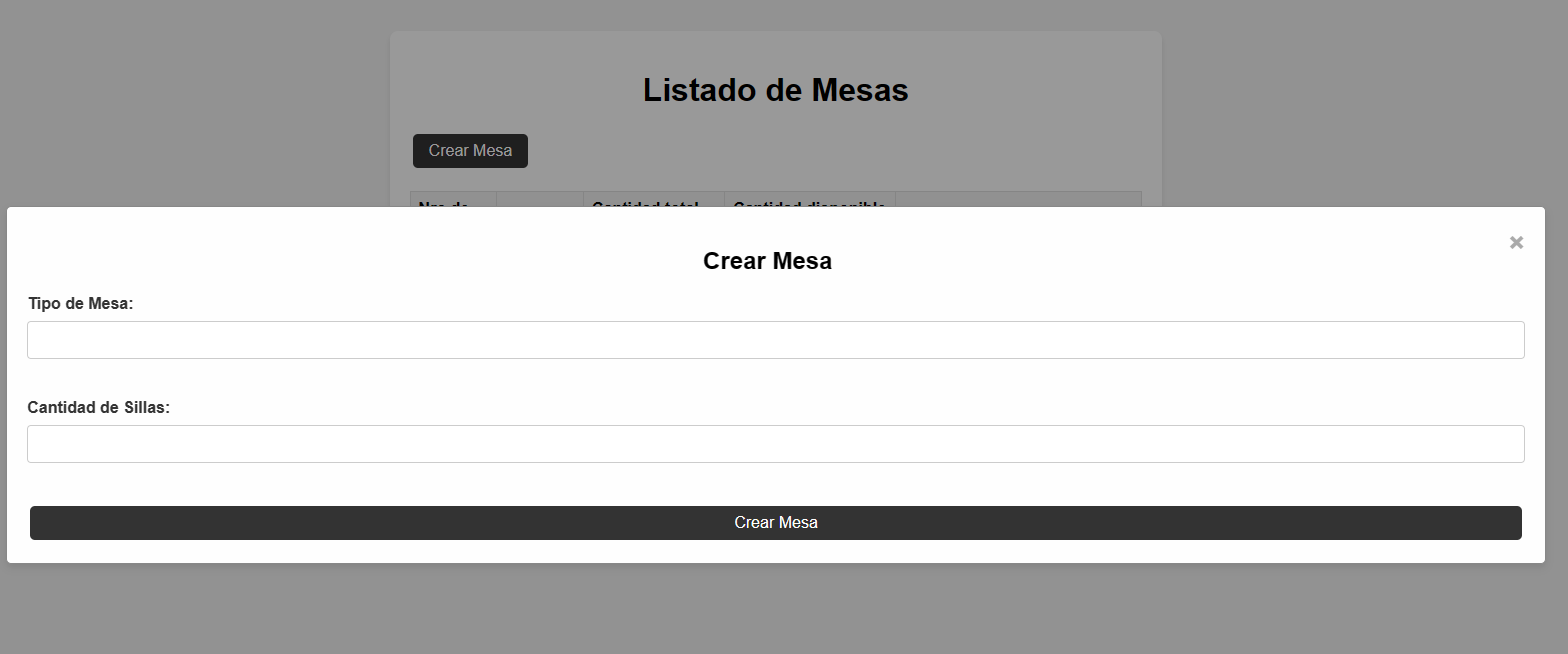


##### Detallar Caso de Uso

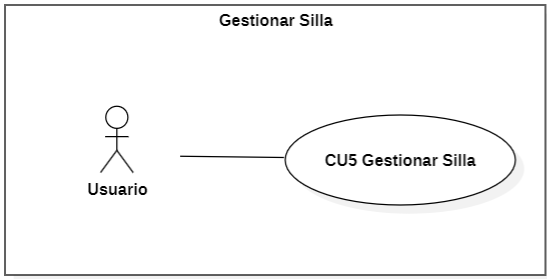
|  |  |
| --- | --- |
| **Nombre del caso de uso** | CU4. Gestionar Mesa |
| **Propósito** | Permite gestionar las mesas de los eventos creados por el administrador. |
| **Actores** | Usuario |
| **Flujo Principal** | **Respuesta del Sistema** |
| 1. **Listar Mesas** 2. Click en “Ver Mesas” del evento deseado. | El sistema obtiene y muestra todas las mesas registradas por el administrador de dicho evento. |
| 1. **Registrar Mesa.** 2. Pulsar el botón “Crear Mesa”. 3. Rellenar el formulario. 4. Pulsar el botón “Crear Mesa”. | El sistema crea la mesa con los datos proporcionados. |
| 1. **Editar Mesa.** 2. Pulsar el botón “Editar” de la mesa deseada. 3. Actualizar los datos de la mesa. 4. Pulsar el botón “Guardar Cambios”. | El sistema actualiza los datos de la mesa. |
| 1. **Eliminar Mesa.** 2. Pulsar el botón “Eliminar” de la mesa deseada. 3. Confirmar Eliminación pulsando el botón “Eliminar”. | El sistema elimina la mesa del evento.  El sistema coloca la mesa asignada de la invitación en cero. |
| **Excepciones** | Eliminar una mesa que haga referencia a alguna invitación. |

##### Prototipo de la Interfaz





#### Caso de Uso 5.- Gestionar Silla



##### Detallar Caso de Uso

|  |  |
| --- | --- |
| **Nombre del caso de uso** | CU5. Gestionar Silla |
| **Propósito** | Permite gestionar las sillas de las mesas del evento creado por el administrador. |
| **Actores** | Usuario |
| **Flujo Principal** | **Respuesta del Sistema** |
| 1. **Listar Sillas** 2. Click en “Listar Eventos” en el menú de navegación. 3. Click en “Ver Mesas” del evento deseado. | El sistema obtiene y muestra todas las mesas registradas por el administrador de dicho evento. |
| 1. **Editar Sillas.** 2. Pulsar el botón “Agregar Sillas”. 3. Indicar el número de sillas para la mesa. 4. Pulsar el botón “Actualizar sillas”. | El sistema actualiza el número de sillas de la mesa. |
| **Excepciones** | Editar menor cantidad de sillas en la mesa que invitaciones para dicha mesa. |

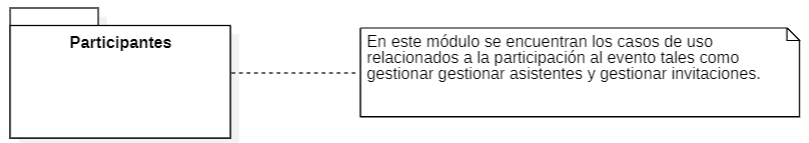
##### Prototipo de la Interfaz



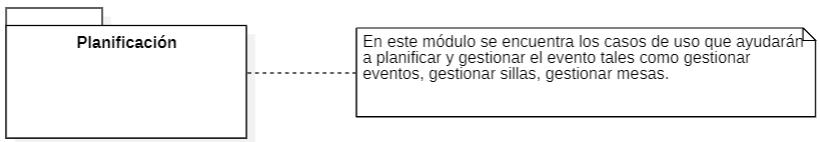
## 2. FLUJO: ANALISIS

### IDENTIFICACION DE LOS MODULOS

#### Módulo de Participantes

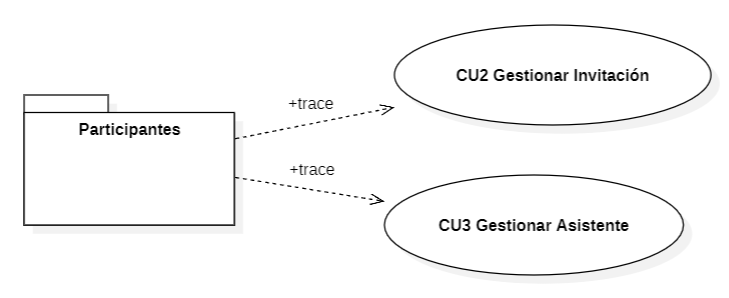


#### Módulo de Planificación

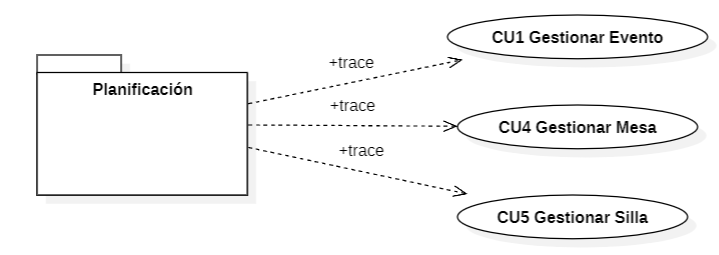


### VISTA DE LOS MODULOS

#### Vista del módulo de Participantes



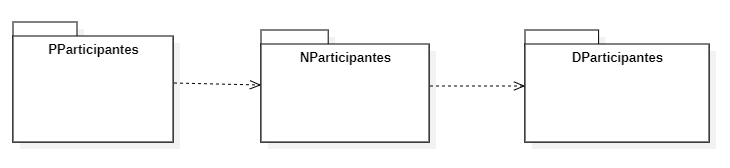
#### Vista del módulo de Planificación



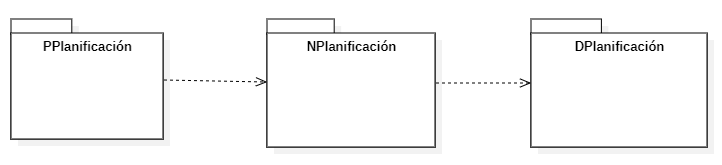
## FLUJO: DISEÑO

### DISEÑO DE LA ARQUITECTURA DE SOFTWARE

#### Módulo de Participantes

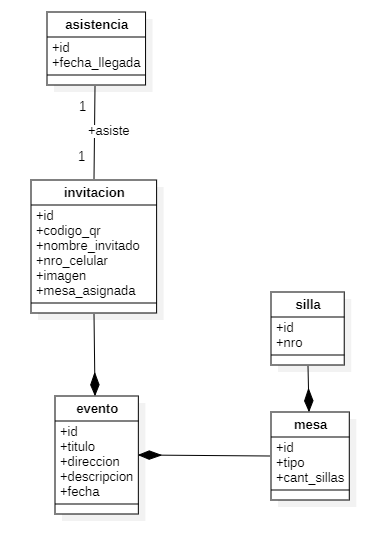


#### Módulo de Planificación



### DISEÑO DE LA BASE DE DATOS

#### Diseño Conceptual de la Base de Datos



#### Diseño Lógico de la Base de Datos



#### Diseño Físico de la Base de Datos

create table evento(

id serial primary key,

titulo varchar(30),

direccion varchar(40),

descripcion varchar,

fecha varchar(30),

);

create table mesa(

id serial primary key,

tipo varchar(20),

capacidad smallint,

id\_evento int,

foreign key(id\_evento) references evento(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

create table silla(

id serial primary key,

nro smallint,

id\_mesa int,

foreign key(id\_mesa) references mesa(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

create table invitacion(

id serial primary key,

nombre\_invitado varchar,

nro\_celular int,

mesa\_asignada int,

id\_evento int,

foreign key(id\_evento) references evento(id)

ON UPDATE CASCADE

ON DELETE CASCADE,

);

create table asistencia(

id serial primary key,

fecha\_llegada varchar(20),

id\_invitacion int,

foreign key(id\_invitacion) references invitacion(id)

ON UPDATE CASCADE

ON DELETE CASCADE

);

### DISEÑO DE LA INTERFAZ

### DISEÑO DE DETALLE PROCEDIMENTAL

### DIAGRAMA DE SECUENCIA

#### CU1. Gestionar Evento

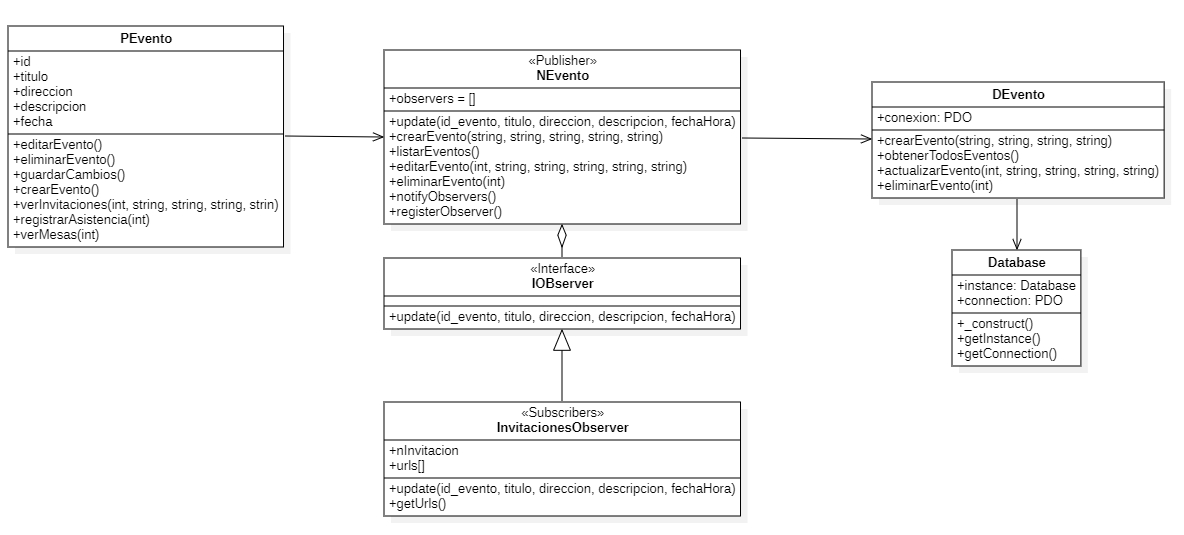
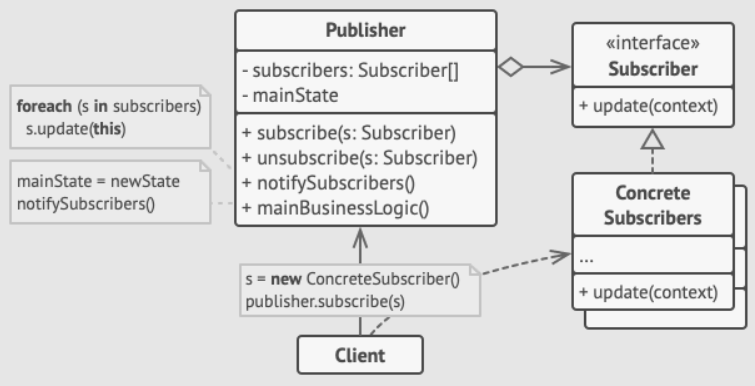
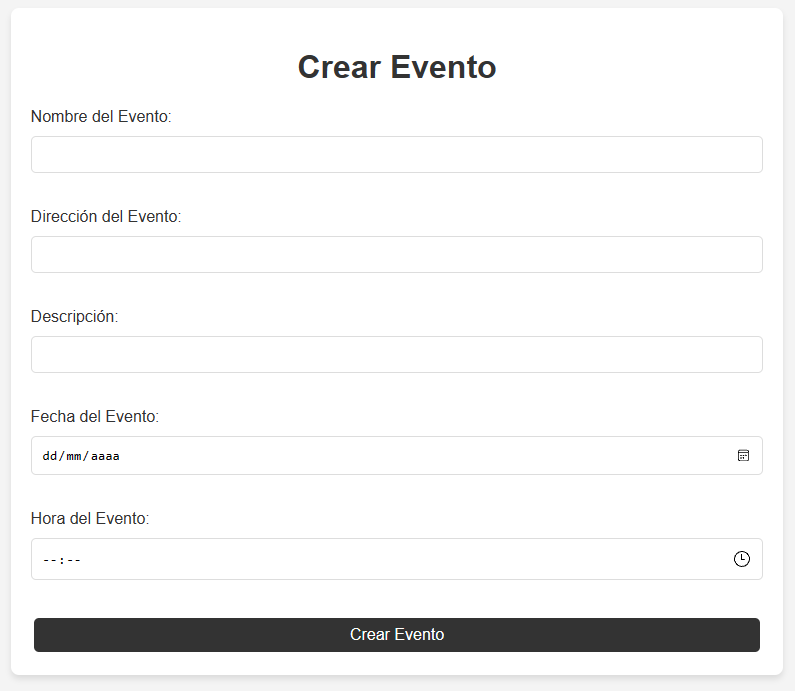


Diagrama genérico



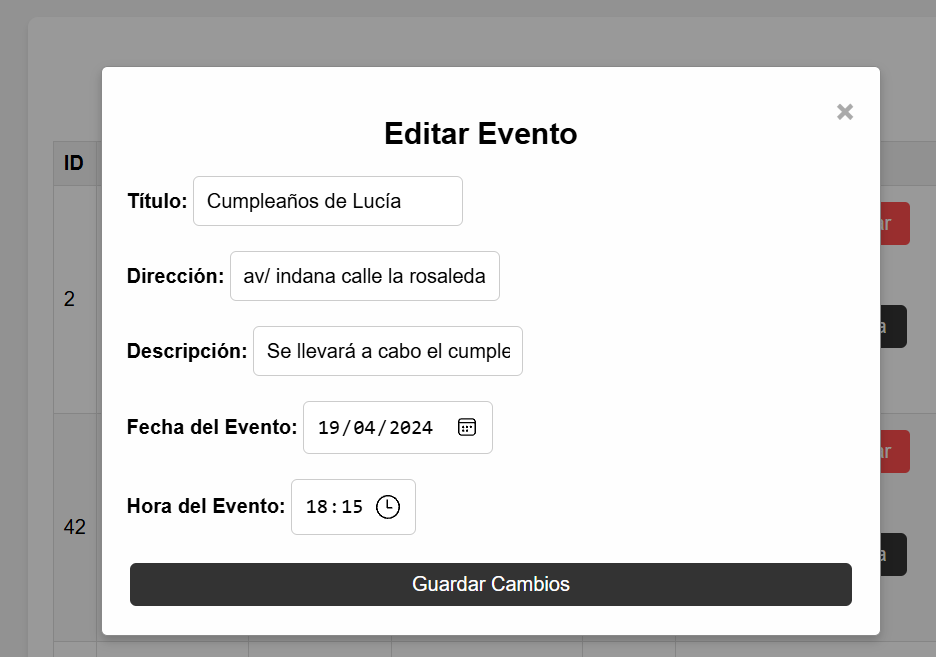
* Interfaz Crear Evento



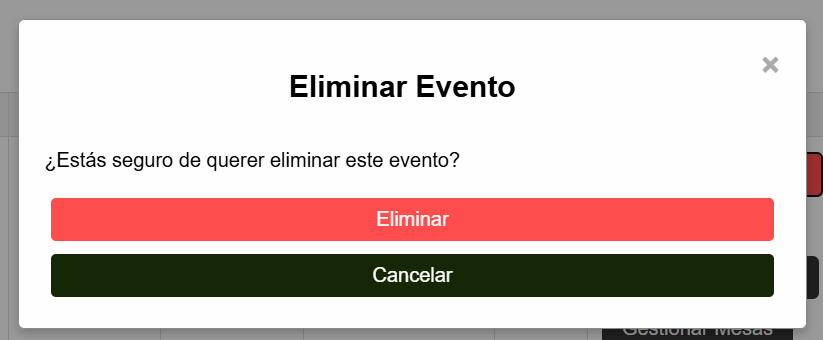
* Lista Evento

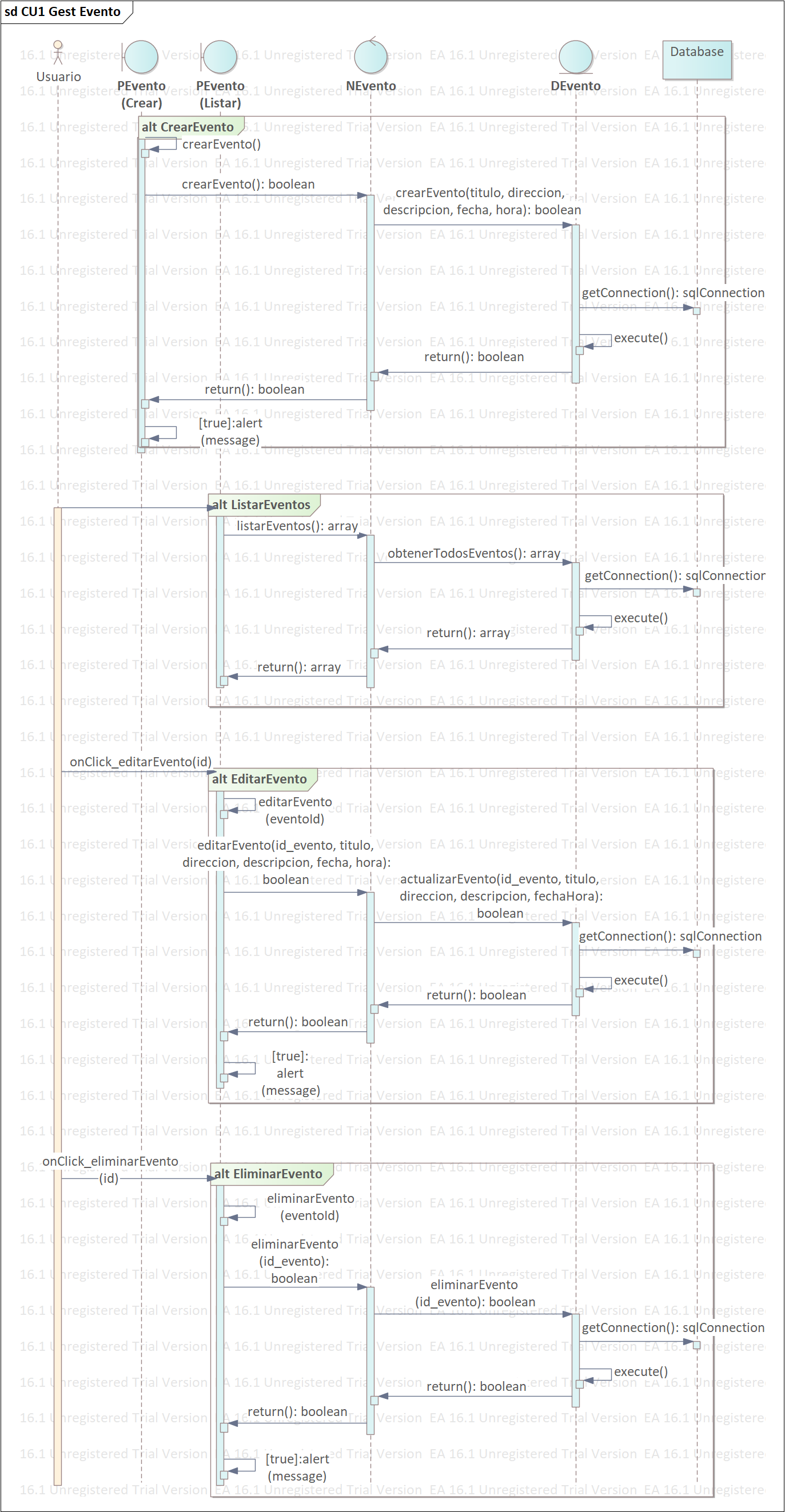


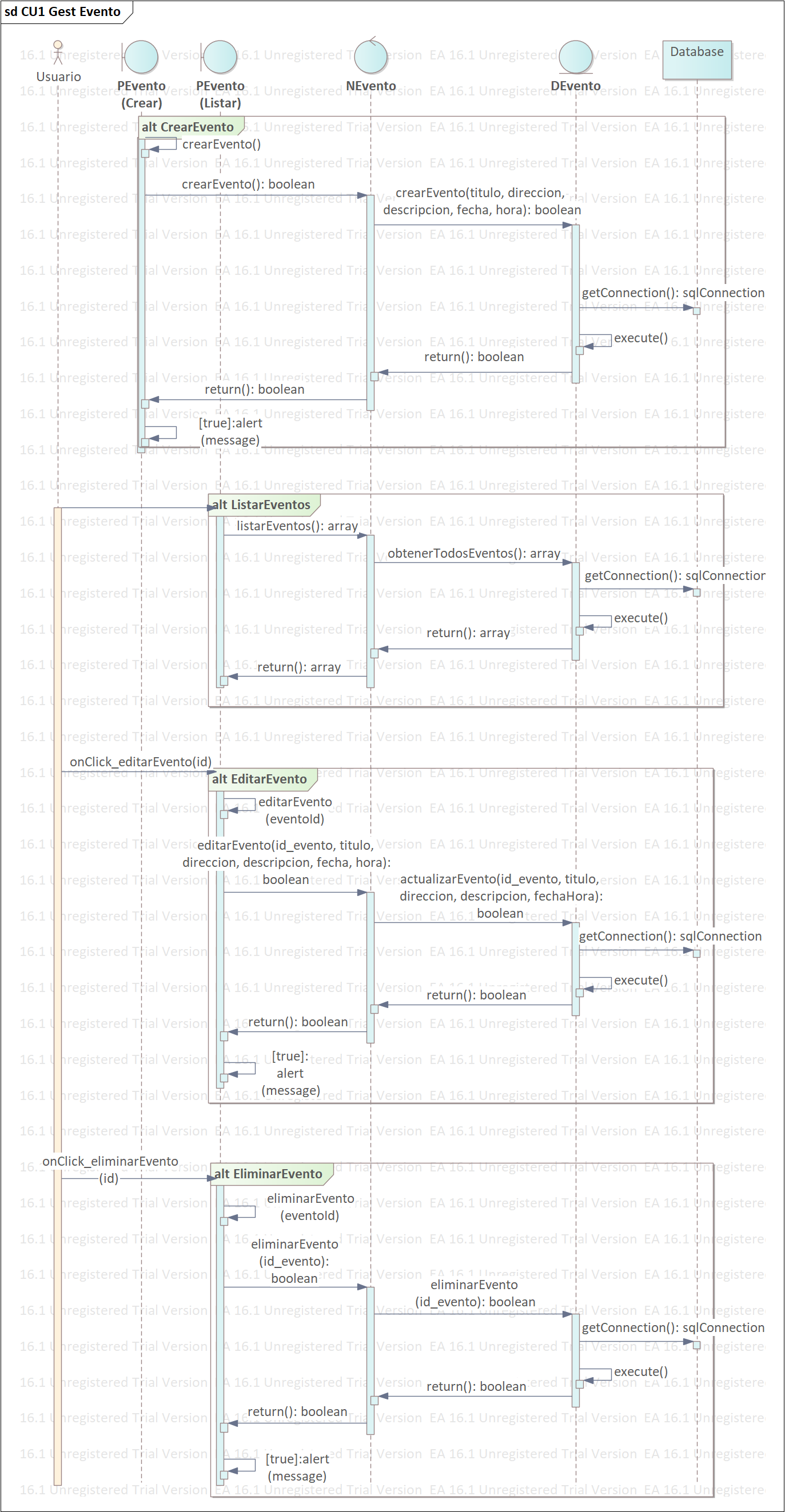
* Editar Evento

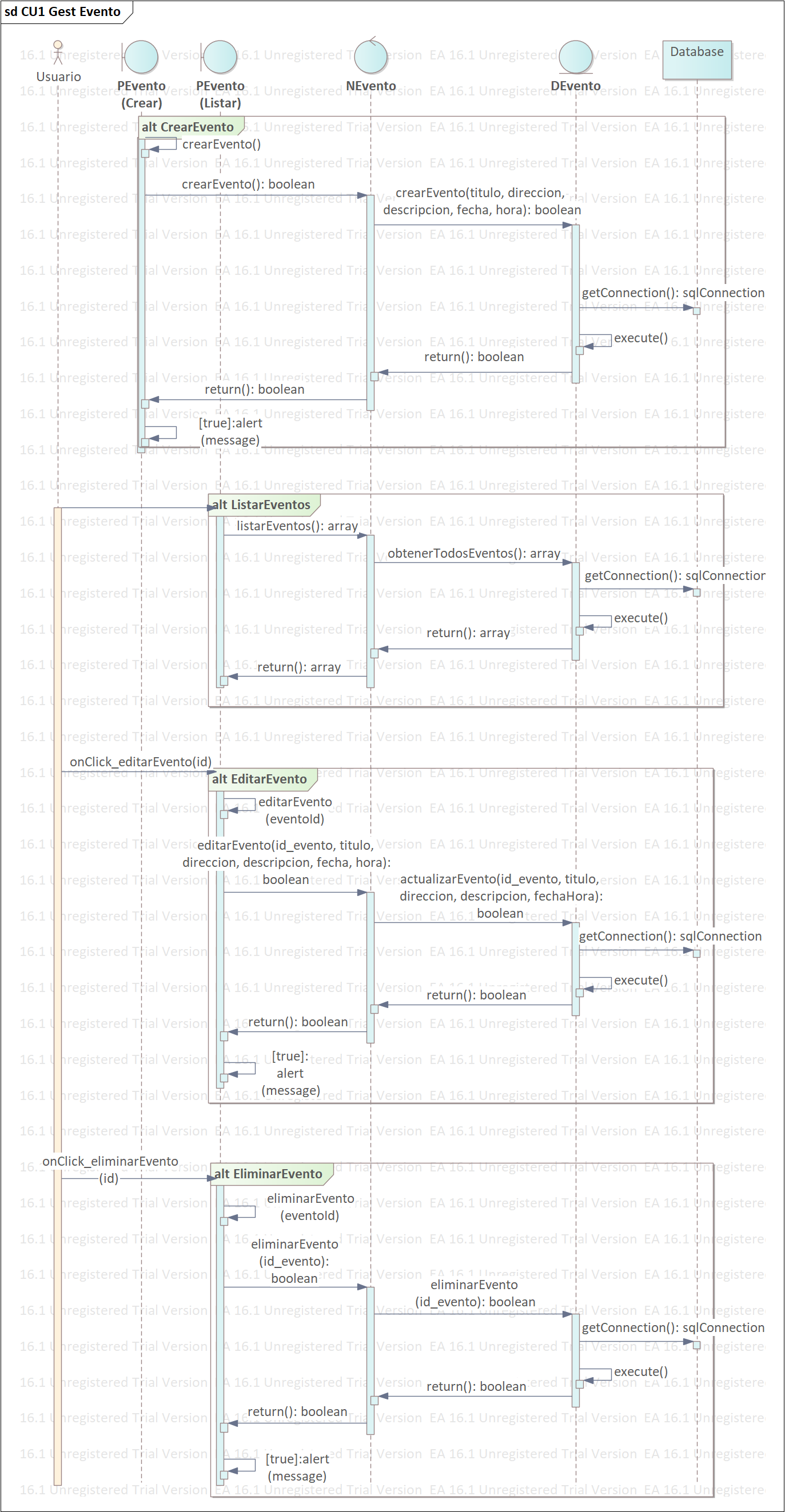


* Eliminar Evento



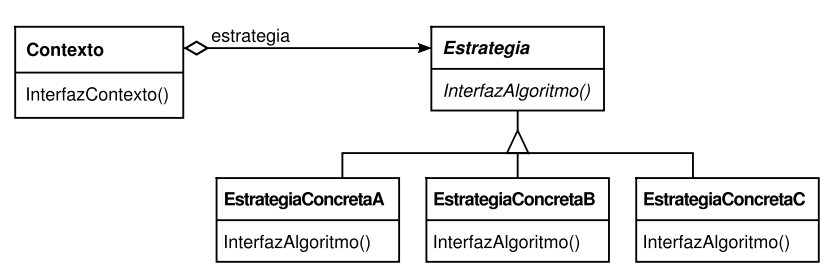






#### CU2. Gestionar Invitación

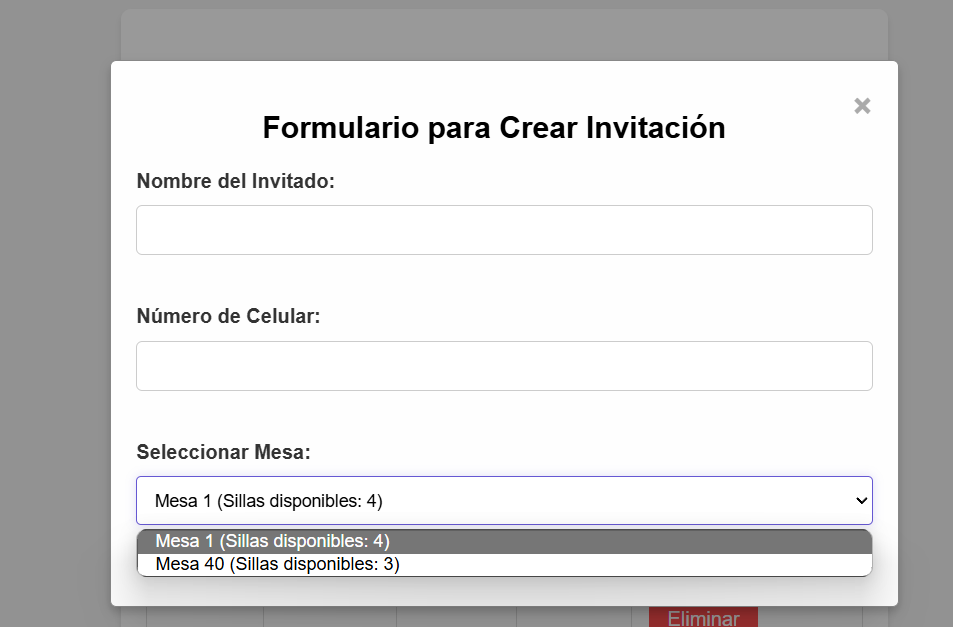
Diagrama Genérico



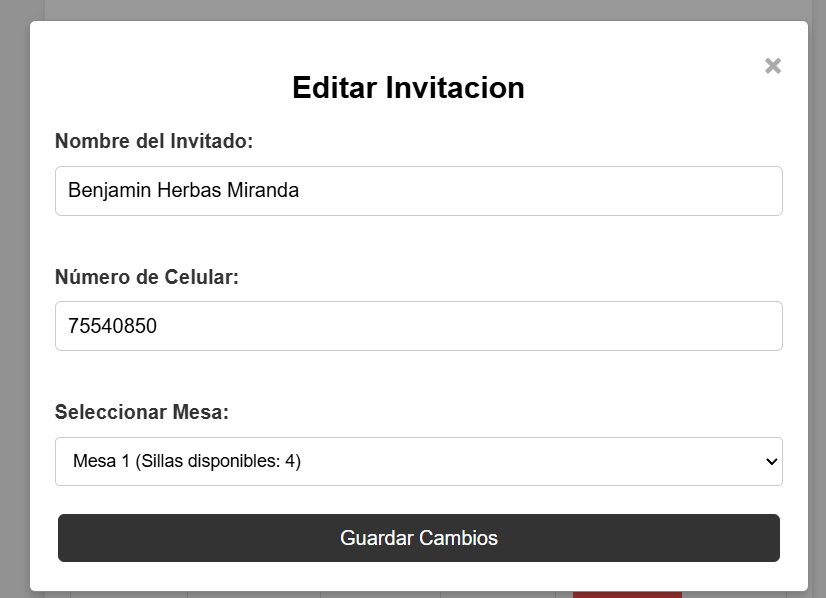
* Lista Invitaciones de un Evento



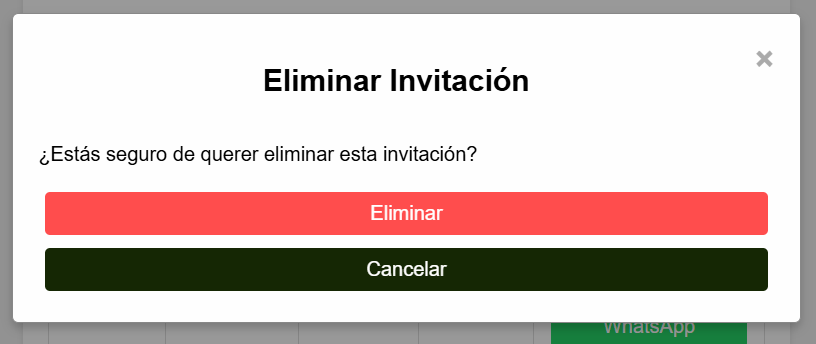
* Crear Invitación



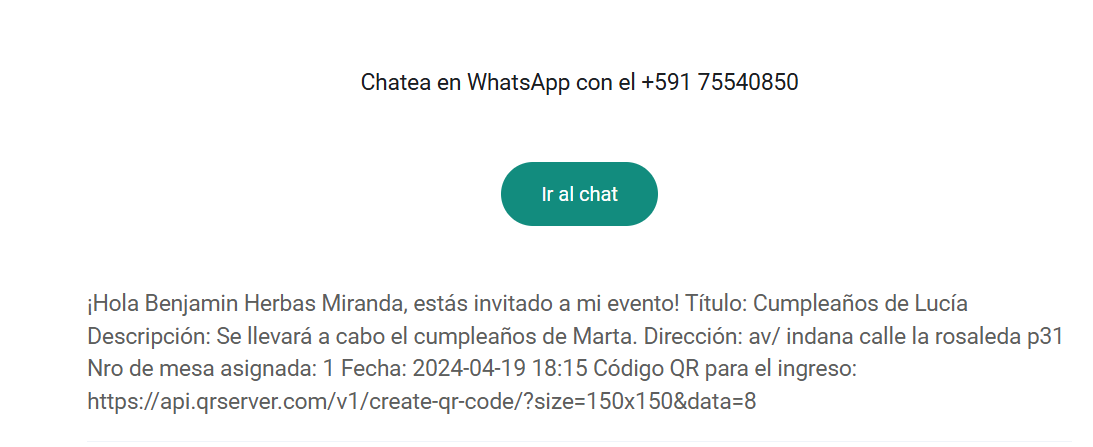
* Editar Invitación



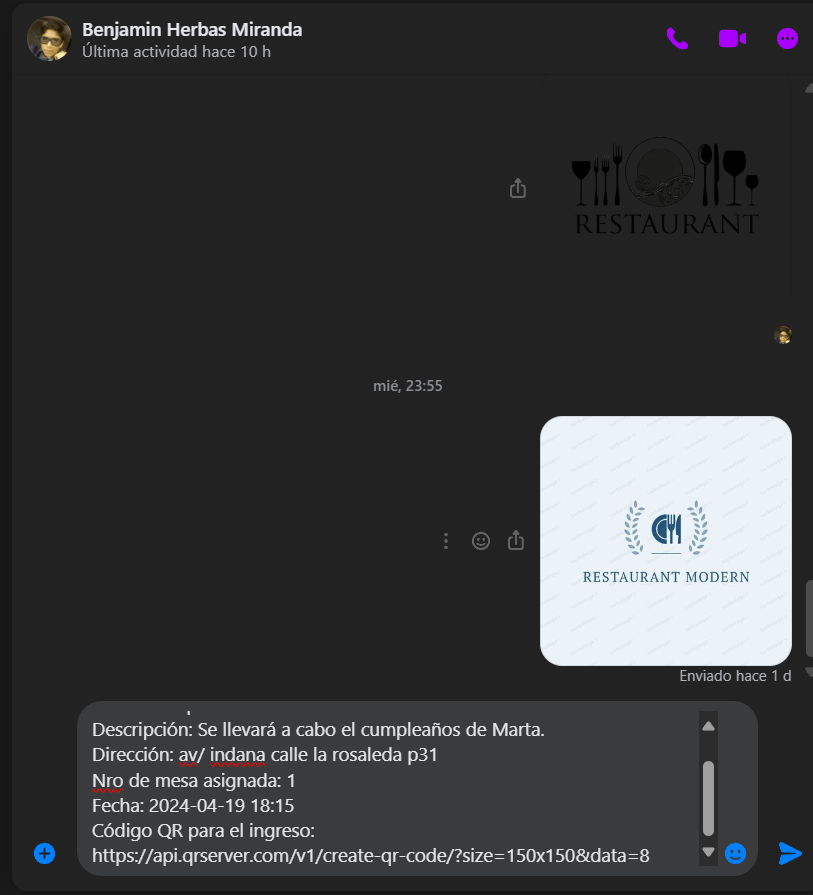
* Eliminar Invitación



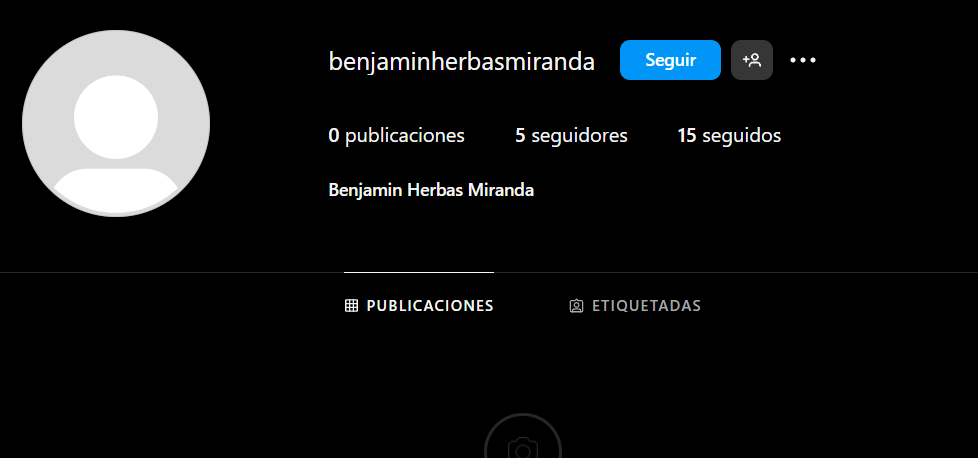
* Compartir Invitación por WhatsApp

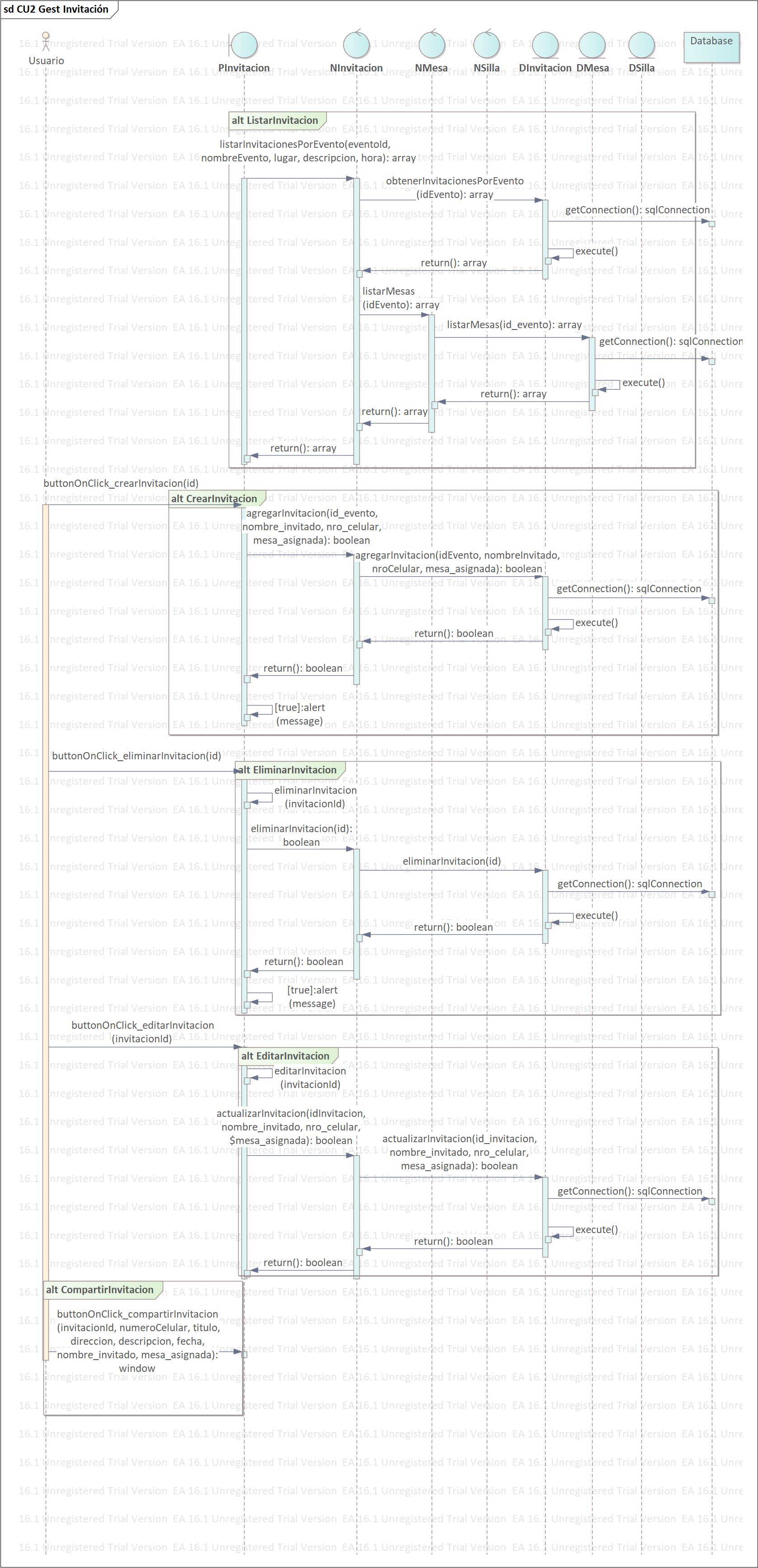


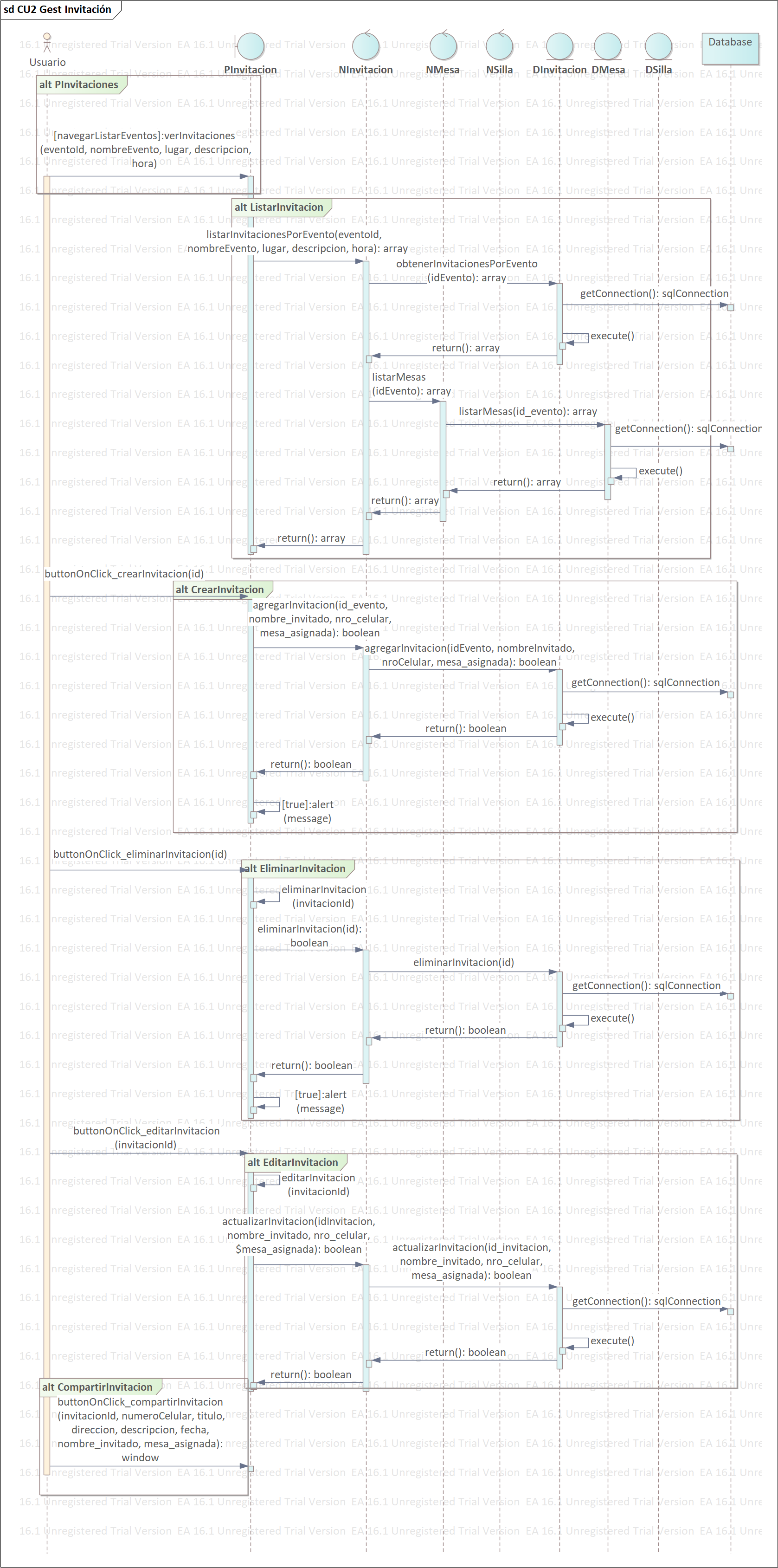
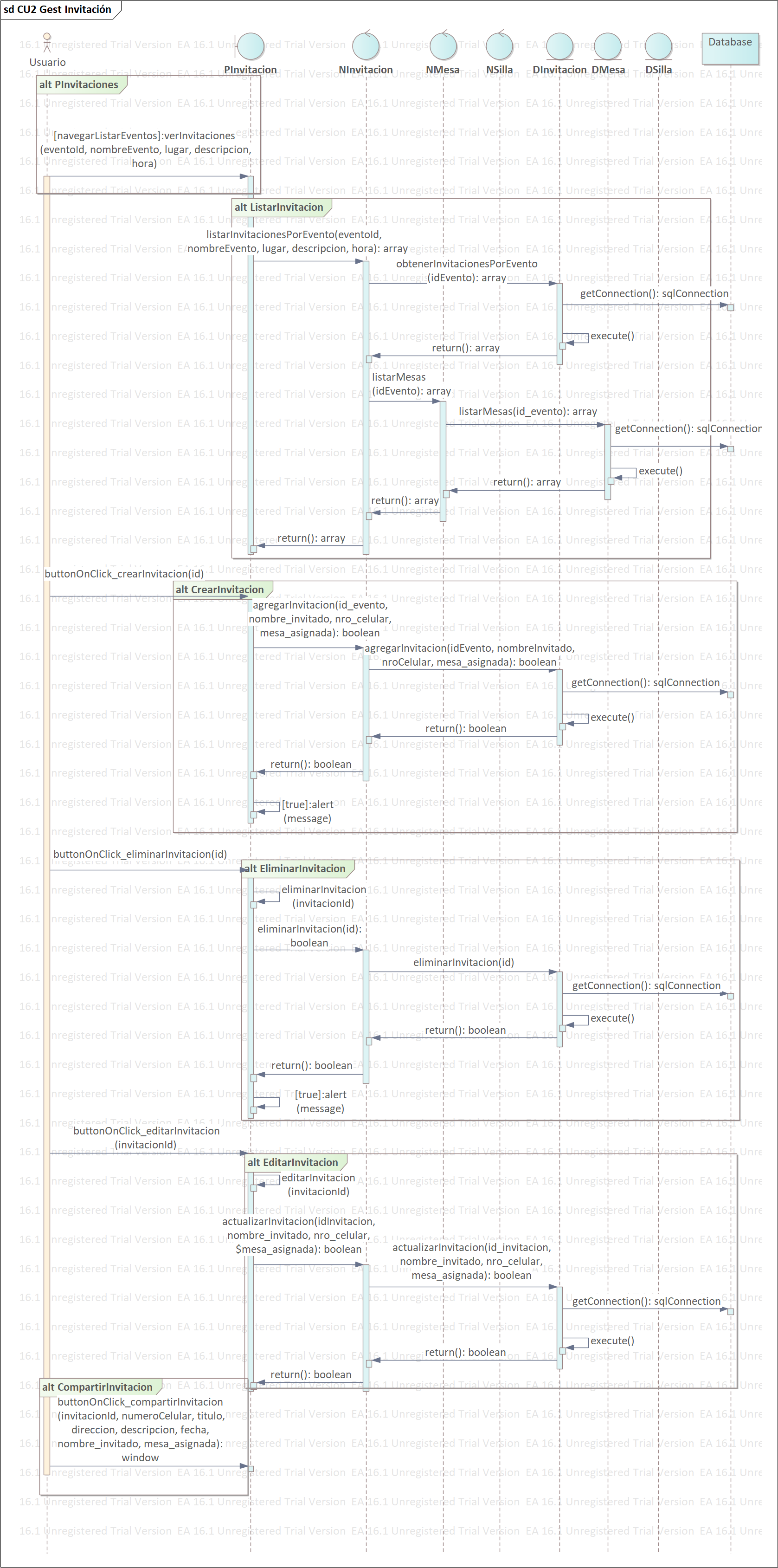
* Compartir Invitación por Facebook



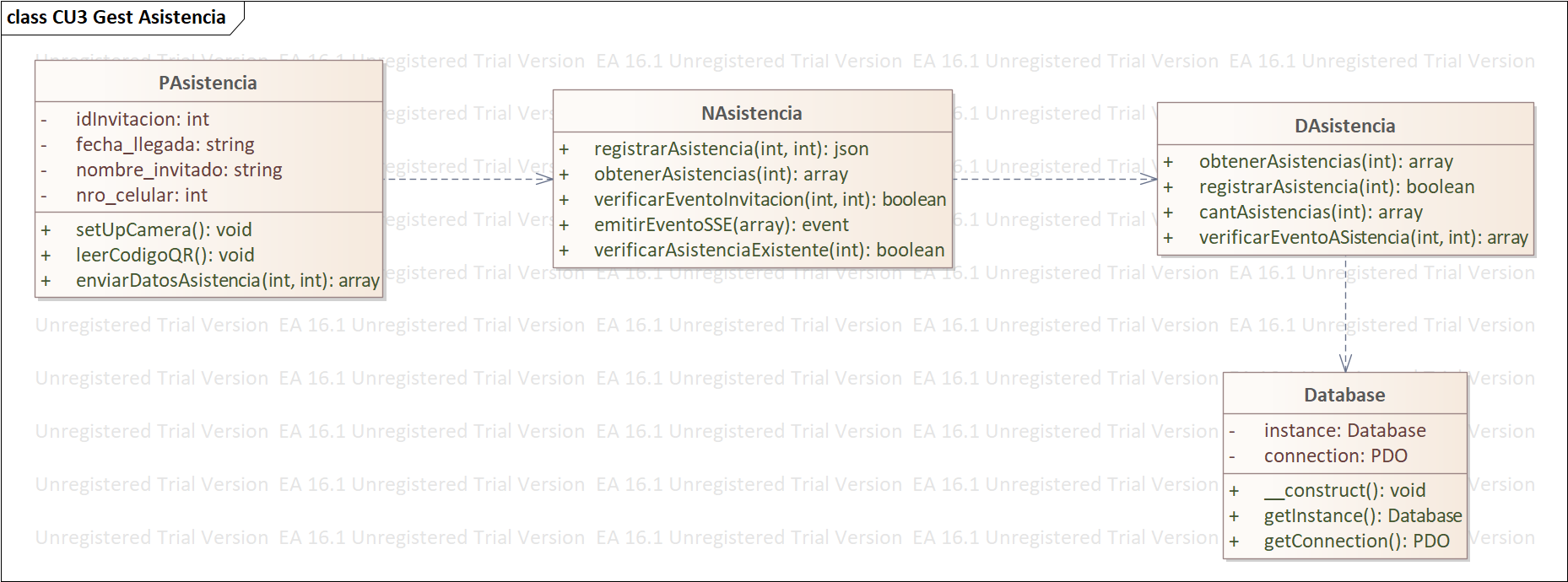
* Compartir Invitación por Instagram

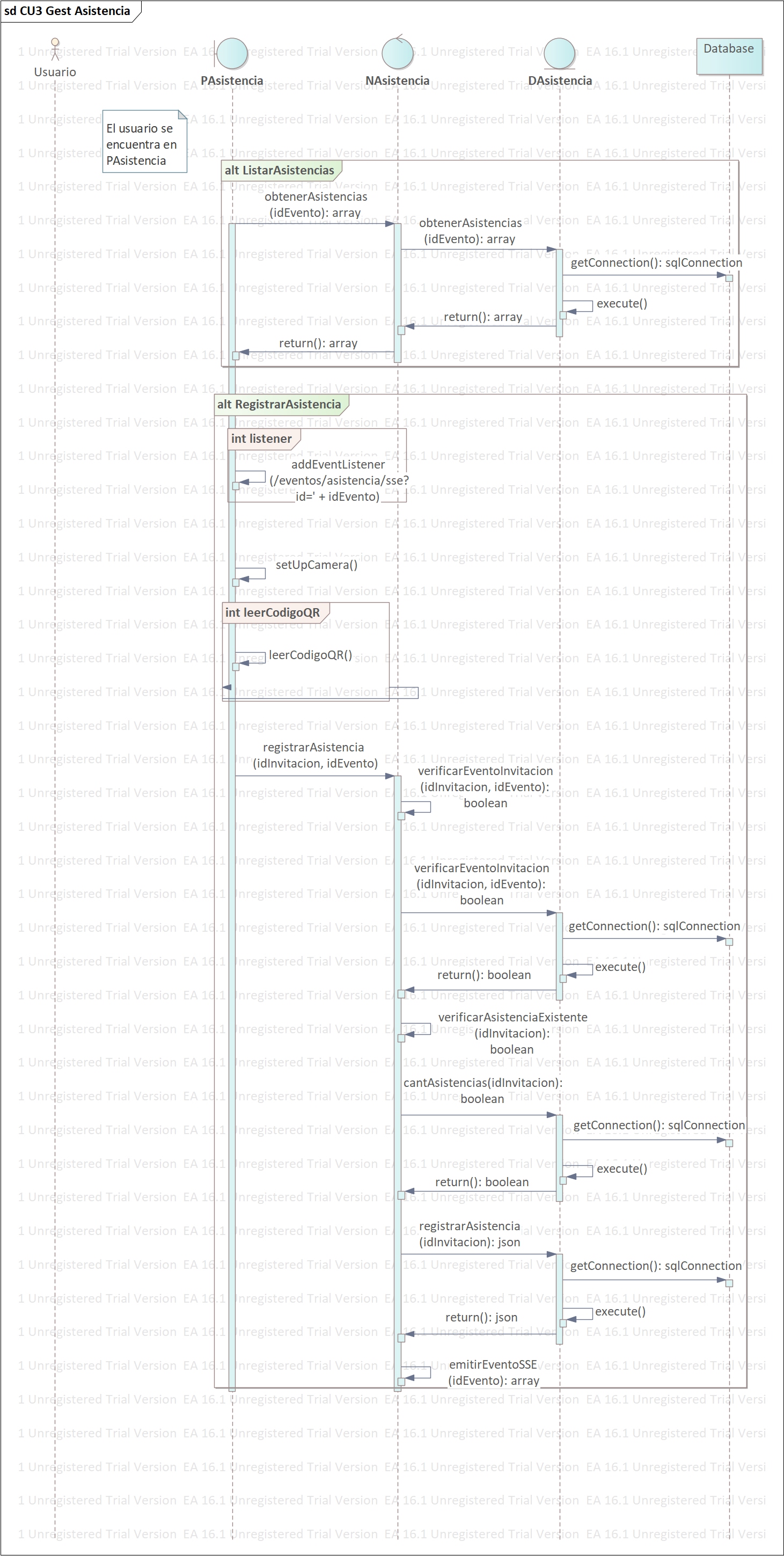


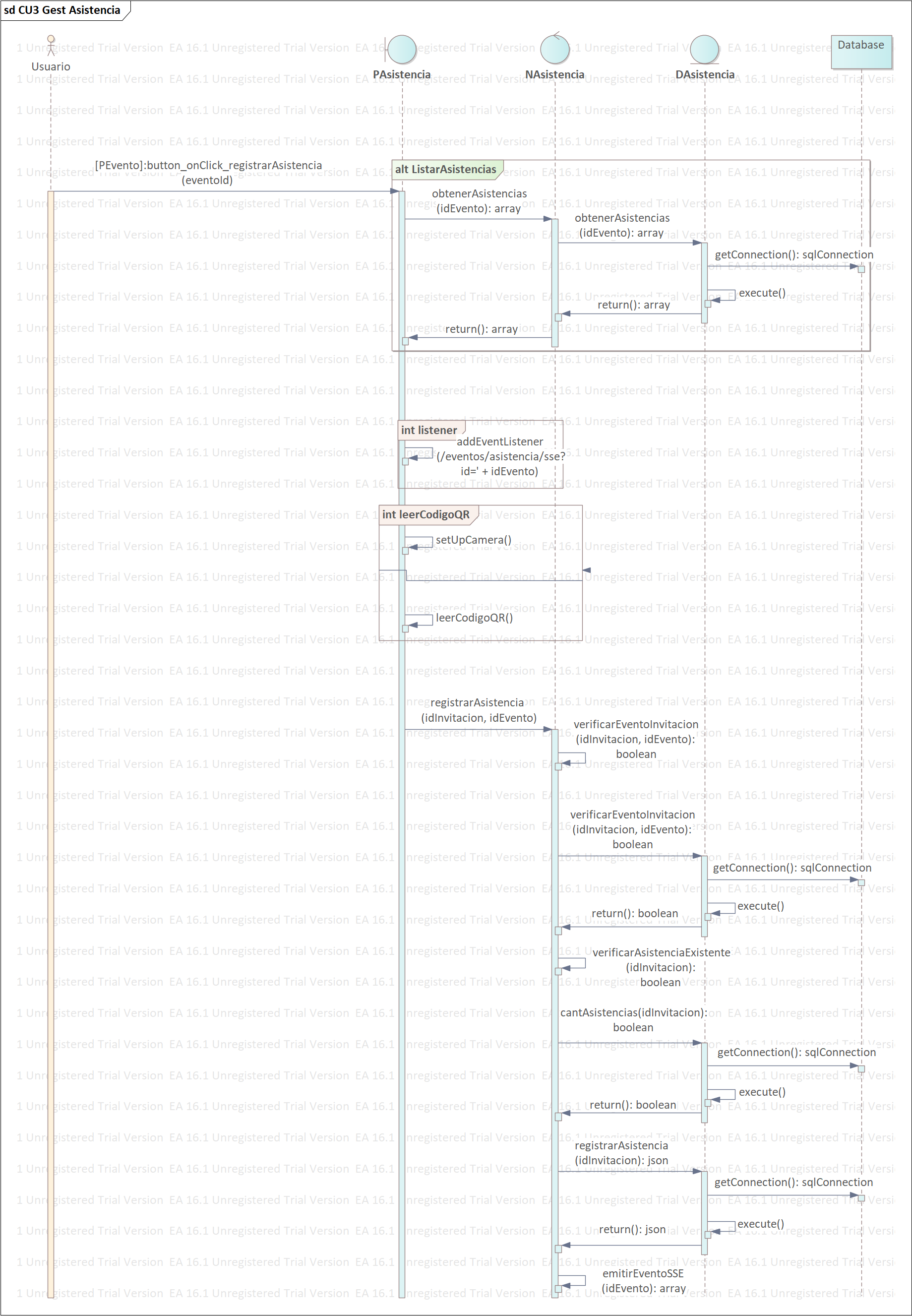


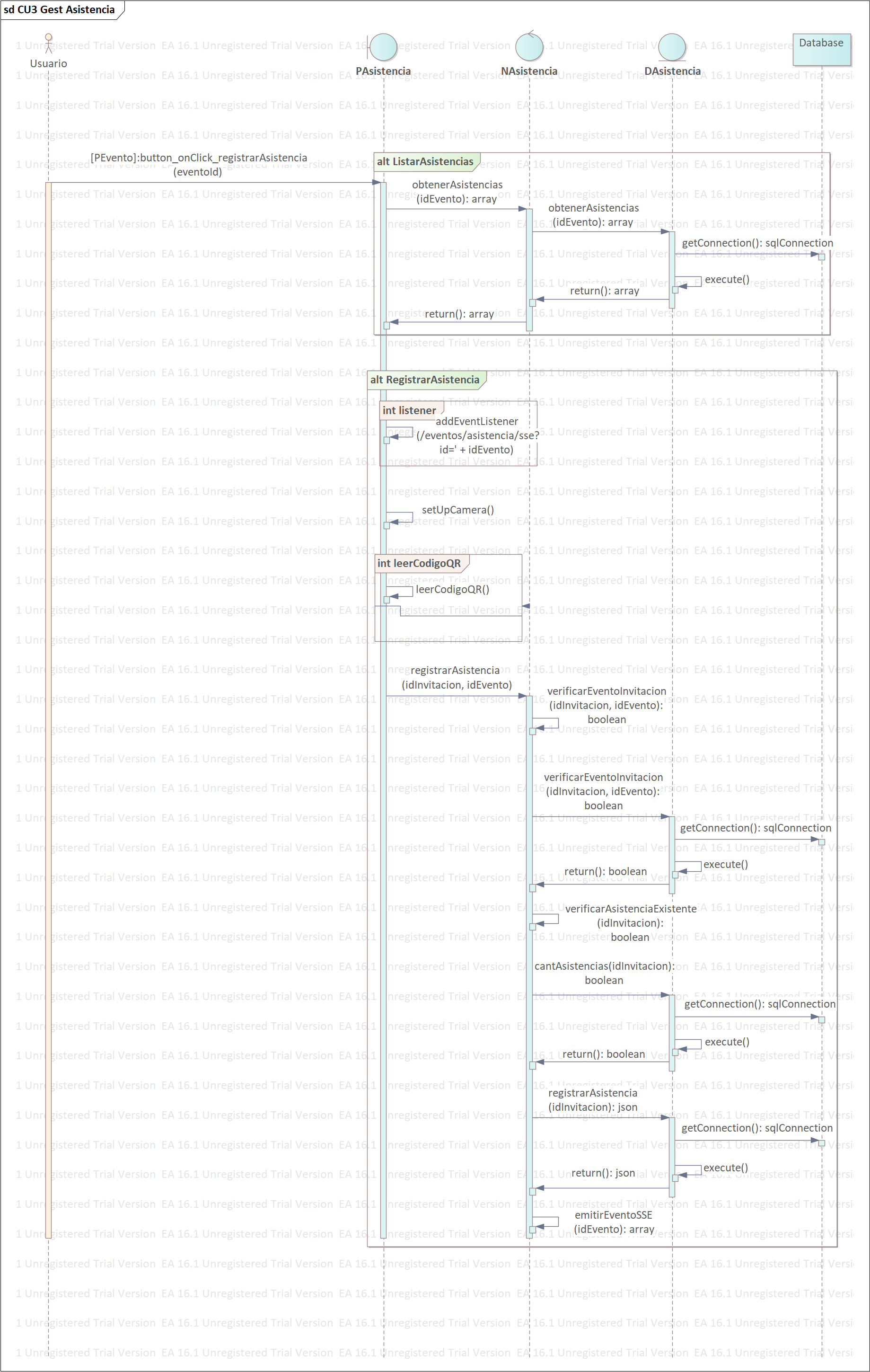


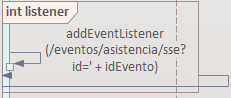
#### CU3. Gestionar Asistencia









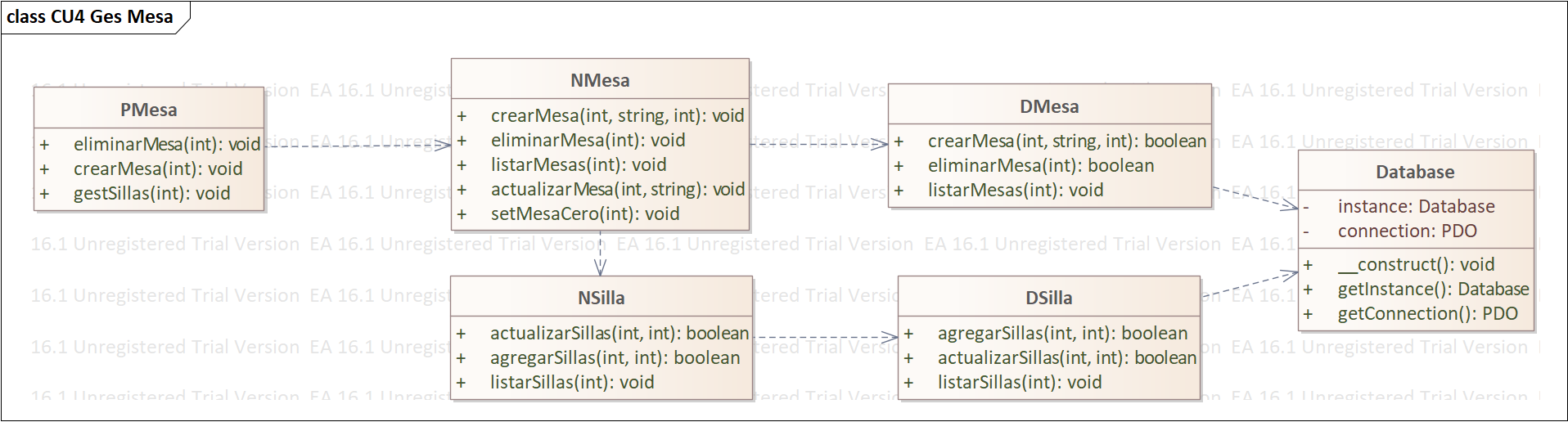


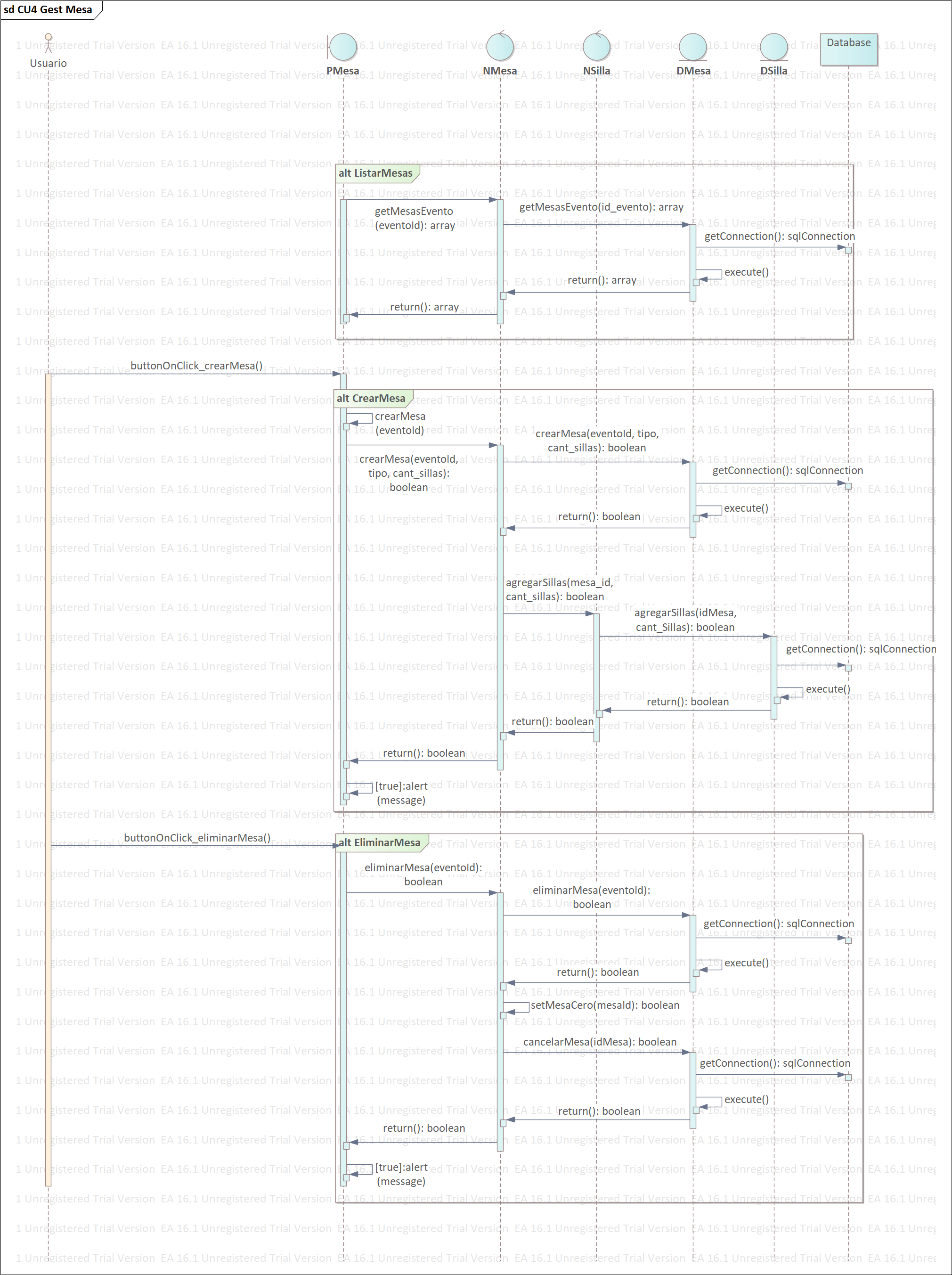


* Registro de Asistencias y Listado de Asistentes



#### CU4 Gestionar Mesa

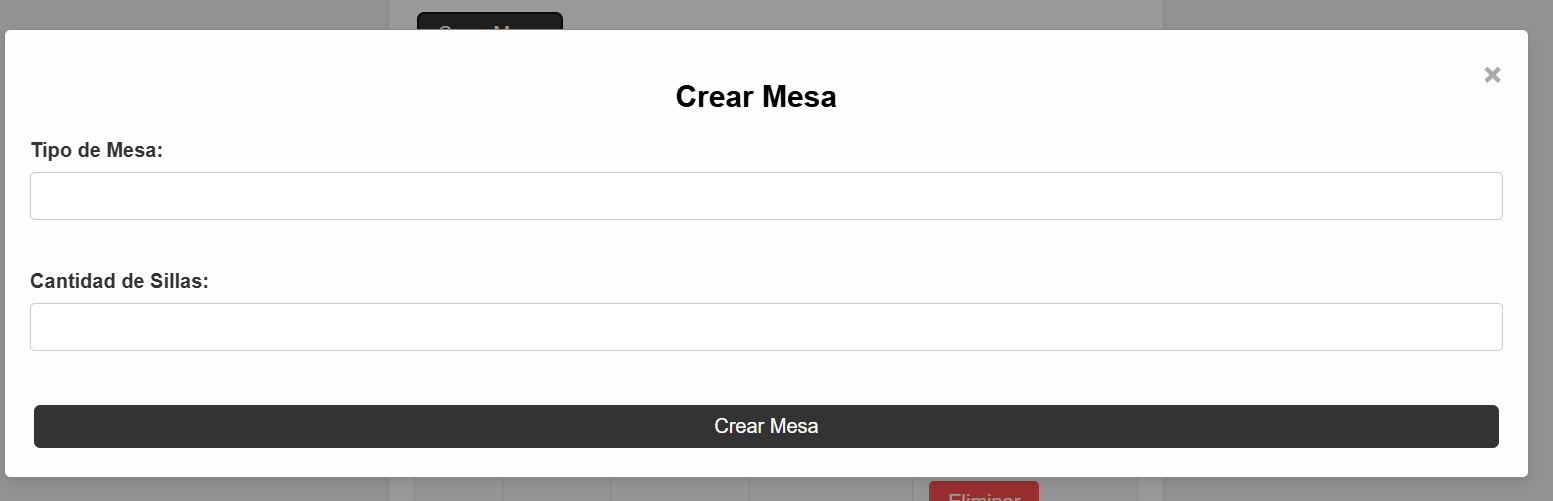




* Listado de Mesas



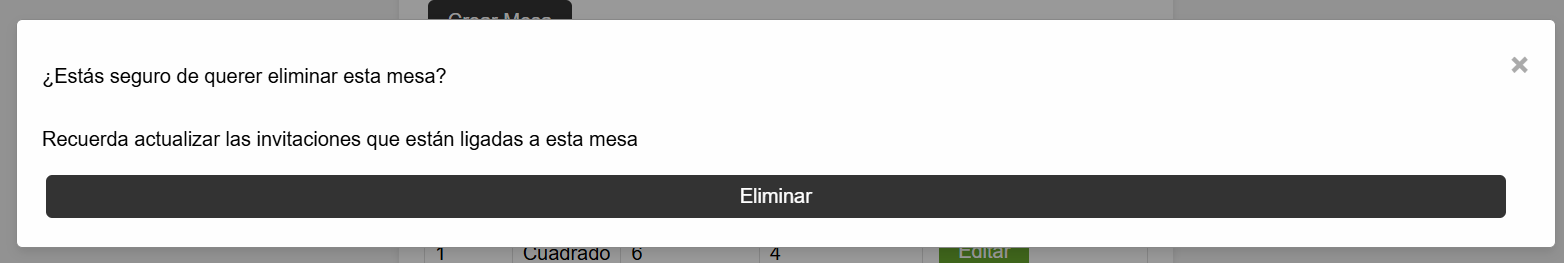
* Crear Mesa



* Editar Mesa

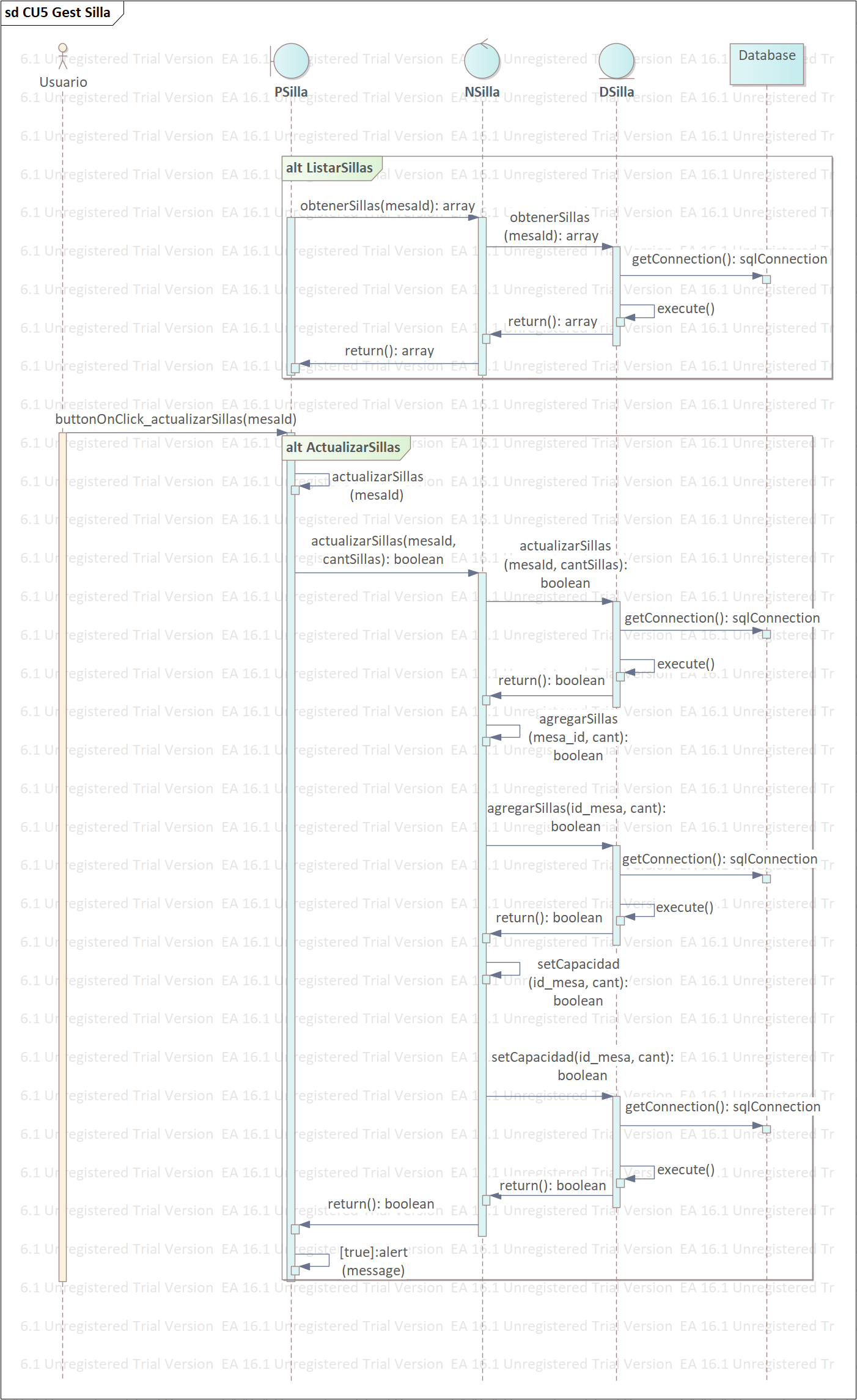


* Eliminar Mesa



#### CU5 Gestionar Silla





* Listado de Sillas



* Editar Sillas de la Mesa



## CODIGO

### CU1. GESTIONAR EVENTO

* Capa de Presentación
  + [RegEvent/src/Presentacion/views/eventos at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/tree/main/src/Presentacion/views/eventos)
* Capa de Negocio
  + [RegEvent/src/Negocio/NEvento.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Negocio/NEvento.php)
* Capa de Dato
  + [RegEvent/src/Datos/DEvento.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Datos/DEvento.php)

### CU2. GESTIONAR INVITACION

* Capa de Presentación
  + [RegEvent/src/Presentacion/views/invitaciones/listar.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Presentacion/views/invitaciones/listar.php)
* Capa de Negocio
  + [RegEvent/src/Negocio/NInvitacion.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Negocio/NInvitacion.php)
* Capa de Dato
  + [RegEvent/src/Datos/DInvitacion.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Datos/DInvitacion.php)

### CU3. GESTIONAR ASISTENCIA

* Capa de Presentación
  + [RegEvent/src/Presentacion/views/Asistencia/registrar.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Presentacion/views/Asistencia/registrar.php)
* Capa de Negocio
  + [RegEvent/src/Negocio/NAsistencia.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Negocio/NAsistencia.php)
* Capa de Dato
  + [RegEvent/src/Datos/DAsistencia.PHP at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Datos/DAsistencia.PHP)

### CU4. GESTIONAR MESA

* Capa de Presentación
  + [RegEvent/src/Presentacion/views/mesas/gestionar.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Presentacion/views/mesas/gestionar.php)
* Capa de Negocio
  + [RegEvent/src/Negocio/NMesa.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Negocio/NMesa.php)
* Capa de Dato
  + [RegEvent/src/Datos/DMesa.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Datos/DMesa.php)

### CU5. GESTIONAR SILLA

* Capa de Presentación
  + [RegEvent/src/Presentacion/views/sillas/gestionar.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Presentacion/views/sillas/gestionar.php)
* Capa de Negocio
  + [RegEvent/src/Negocio/NSilla.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Negocio/NSilla.php)
* Capa de Dato
  + [RegEvent/src/Datos/DSilla.php at main · mariobros123413/RegEvent (github.com)](https://github.com/mariobros123413/RegEvent/blob/main/src/Datos/DSilla.php)

## BIBLIOGRAFIA

Gonzalez, D. A. (s.f.). *Pases QR*. Obtenido de https://pasesqr.com/index.html

John, E. G. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software.* XcUiDi. Obtenido de https://virtual.unju.edu.ar/pluginfile.php/576599/mod\_resource/content/1/Patrones%20de%20Diseno%20-%20Erich%20Gamma.pdf

scanapp. (s.f.). *scanapp.org*. Obtenido de https://scanapp.org/html5-qrcode-docs/docs/apis

Zhart, D. (2024). *https://refactoring.guru/es/design-patterns/*. Obtenido de https://refactoring.guru/es/design-patterns/