

Universidad Veracruzana

Facultad de Estadística e Informática
Licenciatura en Ingeniería de Software

Documento de Especificación de Requerimientos
Sistema “Ared Espacio”
100% de avance del Proyecto

Integrantes del equipo

JUAREZ CAPISTRAN MAURICIO
CORTINA GAMBOA EHECATL SAID

Contenido

| | |
|----------------------------------------------------|----|
| Descripción de los casos de uso Implementados..... | 4 |
| Diagramas de Robustez..... | 27 |
| CU-01. Registrar Alumno..... | 27 |
| CU-02. Modificar Alumno..... | 27 |
| CU-03. Dar de Baja Alumno..... | 28 |
| CU-04. Registrar Maestro..... | 29 |
| CU-05. Modificar Maestro..... | 29 |
| CU-06. Dar de Baja Maestro..... | 30 |
| CU-07. Crear Grupo..... | 31 |
| CU-08 Modificar Grupo y CU-09 Eliminar Grupo | 31 |
| CU-10. Pagar Maestro | 32 |
| CU-11. Registrar Egreso..... | 32 |
| CU-12. Ver historial de Egresos..... | 33 |
| CU-13. Ver historial de Ingresos..... | 33 |
| CU-14. Ver Notificaciones | 34 |
| CU-15. Crear Promoción..... | 34 |
| CU-16. Modificar Promoción..... | 35 |
| CU-17. Eliminar Promoción | 35 |
| CU-18. Registrar Asistencia | 36 |
| CU-19. Cambiar Alumno de Grupo..... | 36 |
| CU-20 Inscribir Alumno | 37 |
| CU-21 Pagar Mensualidad..... | 38 |
| Modelo de Dominio | 39 |
| Diagrama de Clases | 40 |
| Diagramas de Secuencia..... | 41 |
| CU-01 Registrar Alumno..... | 41 |
| CU-02 Modificar Alumno..... | 42 |
| CU-03 Dar de Baja Alumno..... | 43 |
| CU-04 Registrar Maestro..... | 44 |
| CU-05 Modificar Maestro..... | 45 |
| CU-06 Dar de baja Maestro..... | 46 |
| CU-07 Registrar Grupo | 47 |

| | |
|----------------------------------------------------|----|
| CU-08 Modificar Grupo y CU-09 Eliminar Grupo | 48 |
| CU-10. Pagar Maestro | 49 |
| CU-11. Registrar Egreso..... | 50 |
| CU-12. Ver historial de Egresos | 51 |
| CU-13. Ver historial de Ingresos..... | 52 |
| CU-14. Ver Notificaciones | 53 |
| CU-15. Crear Promoción..... | 54 |
| CU-16. Modificar Promoción..... | 55 |
| CU-17. Eliminar Promoción | 56 |
| CU-18. Registrar Asistencia | 57 |
| CU-19. Cambiar Alumno de Grupo..... | 58 |
| CU-20. Pagar Inscripción | 59 |
| CU-21. Pagar Mensualidad | 60 |
| Modelo Relacional..... | 61 |
| Plan de Pruebas..... | 62 |
| Clase Alumno..... | 62 |
| Clase Maestro..... | 62 |
| Clase Grupo | 65 |
| Clase Foto | 66 |
| Clase Horario | 67 |
| Clase Asistencia | 69 |
| Clase Danza | 69 |
| Clase Clase..... | 72 |
| Clase Egreso | 73 |
| Clase Promoción..... | 74 |

Avance general del Proyecto de Ared Espacio

Primera entrega 40%

- CU-01. Registrar Alumno
- CU-02. Modificar Alumno
- CU-03. Dar de baja Alumno
- CU-04. Registra Maestro
- CU-05. Modificar Maestro
- CU-06. Dar de baja Maestro
- CU-07. Crear Grupo
- CU-08. Modificar Grupo
- CU-09. Eliminar Grupo

Segunda entrega 90%

- CU-10. Pagar Maestro
- CU-11. Registrar Egreso
- CU-12. Ver historial de Egresos
- CU-13. Ver historial de Ingresos
- CU-14. Ver Notificaciones
- CU-15. Crear Promoción
- CU-16. Modificar Promoción
- CU-17. Eliminar Promoción
- CU-18. Registrar Asistencia
- CU-19. Cambiar Alumno de Grupo

Entrega final 100%

- CU-20. Pagar Inscripción
- CU-21. Pagar Mensualidad

Descripción de los casos de uso Implementados

| | |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-01 |
| Nombre: | Registrar Alumno |
| Autor: | Jonathan Garcia Delgado |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | El sistema deberá permitir a la directora, registrar los nuevos alumnos de la escuela |
| Precondiciones: | PRE-1 El alumno ha pagado inscripción PRE-2 El alumno no ha sido registrado |
| Flujo Normal: | <ol style="list-style-type: none"> 1. La directora selecciona desde el menú inscribir nuevo alumno 2. La directora captura el 'Nombre', 'Apellidos', 'Fecha de Nacimiento', 'Genero', 'Correo Electrónico', 'Teléfono', 'Teléfono Móvil', 'Domicilio' y selecciona el 'Estado', 'Ciudad', 'Código Postal' del alumno 3. La directora da clic en 'Guardar' 4. El sistema registra la 'Fecha de registro' y genera una nueva 'Matricula' 5. El sistema almacena el registro y despliega un mensaje: "El alumno ha sido registrado" |
| Flujo Alterno: | <p>FA 1 - Campos Inválidos</p> <ol style="list-style-type: none"> 2.1 En caso de que el campo del nuevo registro no ha sido valido entonces, 2.2 El sistema despliega un mensaje de advertencia: "El campo no es correcto" <p>FA 2 – Agregar Foto</p> <ol style="list-style-type: none"> 2.2 En caso de que la directora de clic en 'Agregar foto' entonces, La directora selecciona la foto desde el explorador de archivos |
| Excepción: | <p>EX 1 – Error de conexión</p> <ol style="list-style-type: none"> 5.1 Si el servidor no ha sido conectado entonces, El sistema despliega un mensaje de error: "El servidor no ha sido conectado" |
| Postcondiciones: | <p>POST-1 El estado del alumno ha sido cambiado a inscrito</p> <p>POST-2 El registro ha sido almacenado en la base de datos</p> |

| | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-02 |
| Nombre: | Modificar Alumno |
| Autor: | Jonathan Garcia Delgado |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | El sistema deberá permitir a la directora, actualizar el registro del alumno en caso de modificación |
| Precondiciones: | El alumno ha sido registrado El sistema ha realizado el CU14 Buscar Alumno |
| Flujo Normal: | <ol style="list-style-type: none"> 1. La directora selecciona el alumno y da clic en ‘Modificar Alumno’ 2. El sistema despliega ‘Nombre’, ‘Apellidos’, ‘Fecha de Nacimiento’, ‘Genero’, ‘Correo Electrónico’, ‘Teléfono’, ‘Teléfono Móvil’, ‘Domicilio’, ‘Ciudad’, ‘Código Postal’, ‘Estado’ y ‘Foto’ del alumno 3. La directora selecciona el campo, captura la modificación y da clic en ‘Guardar’ 4. El sistema actualiza el registro y despliega un mensaje: “El alumno ha sido actualizado” |
| Flujo Alternativo: | <p>FA 1 - Campos Inválidos</p> <p>3.1 En caso de que el campo del registro modificado no ha sido valido entonces, El sistema despliega un mensaje de advertencia: “El campo no es correcto”</p> <p>FA 2 – Actualizar Foto</p> <p>3.2 En caso de que la directora de clic en ‘Actualizar foto’ entonces, 3.3La directora selecciona la foto desde el explorador de archivos y da clic en ‘Establecer foto’</p> |
| Excepción: | <p>EX 1 – Error de conexión</p> <p>4.1 Si el servidor no ha sido conectado entonces, El sistema despliega un mensaje de error: “El servidor no ha sido conectado”</p> |
| Postcondiciones: | POST-1 El registro del alumno ha sido actualizado en la base de datos |

| | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-03 |
| Nombre: | Dar de baja Alumno |
| Autor: | Jonathan Garcia Delgado |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | El sistema deberá permitir a la directora, inhabilitar el registro del alumno en caso de baja temporal o definitiva |
| Precondiciones: | PRE-1 El alumno ha sido registrado PRE-2 El sistema ha realizado el CU14 Buscar Alumno PRE-3 El alumno no adeuda pagos |
| Flujo Normal: | 1. La directora selecciona el alumno y da clic en 'Baja' 2. El sistema despliega 'Fecha de ingreso', 'Matricula', 'Foto', 'Nombre Completo', 'Correo Electrónico' y 'Teléfonos' del alumno 3. La directora captura el motivo de la baja y da clic en 'Dar de baja' 4. El sistema despliega una pantalla de confirmación e inhabilita el registro 5. El sistema despliega un mensaje: "El alumno ha sido dado de baja" |
| Flujo Alternativo: | FA 1 – Cancelar Baja 4.1 En caso de que la directora de clic en 'No' entonces, El sistema ejecuta el caso de uso CU14 Buscar Alumnos |
| Excepción: | EX 1 – Error de conexión 4.1 Si el servidor no ha sido conectado entonces, El sistema despliega un mensaje de error: "El servidor no ha sido conectado" |
| Postcondiciones: | POST-1 El estado del alumno ha sido cambiado a baja POST-2 El registro del alumno ha sido actualizado en la base de datos |

| | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-04 |
| Nombre: | Registrar Maestro |
| Autor: | Jonathan Garcia Delgado |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | El sistema deberá permitir a la directora, registrar los nuevos maestros colaboradores de la escuela |
| Precondiciones: | PRE-1 El colaborador no ha sido registrado |
| Flujo Normal: | <ol style="list-style-type: none"> 1. La directora selecciona desde el menú ‘Registrar nuevo colaborador’ 2. La directora captura el 'Nombre', 'Apellidos', ‘Fecha de Nacimiento’, ‘Genero’, 'Correo Electrónico', 'Teléfono', 'Teléfono Móvil', ‘Domicilio’ y selecciona el ‘Estado’, ‘Ciudad’, ‘Código Postal’ del colaborador 3. La directora da clic en ‘Guardar’ 4. El sistema registra la ‘Fecha de registro’ y genera un ‘Numero de Colaborador’ 5. El sistema almacena el registro y despliega un mensaje: “El colaborador ha sido registrado” |
| Flujo Alternativo: | <p>FA 1 - Campos Inválidos</p> <ol style="list-style-type: none"> 2.1 En caso de que el campo del nuevo registro no ha sido valido entonces, 2.2 El sistema despliega un mensaje de advertencia: “El campo no es correcto” <p>FA 2 – Agregar Foto</p> <ol style="list-style-type: none"> 2.2 En caso de que la directora de clic en ‘Agregar foto’ entonces, <p>La directora selecciona la foto desde el explorador de archivos</p> |
| Excepción: | <p>EX 1 – Error de conexión</p> <ol style="list-style-type: none"> 5.1 Si el servidor no ha sido conectado entonces, <p>El sistema despliega un mensaje de error: “El servidor no ha sido conectado”</p> |
| Postcondiciones: | <p>POST-1 El estado del colaborador ha sido cambiado a inscrito</p> <p>POST-2 El registro ha sido almacenado en la base de datos</p> |

| | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-05 |
| Nombre: | Modificar Maestro |
| Autor: | Jonathan Garcia Delgado |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | El sistema deberá permitir a la directora, actualizar el registro del colaborador en caso de modificación |
| Precondiciones: | PRE-1 El maestro ha sido registrado PRE-2 El sistema ha realizado el CU18 Buscar Maestro |
| Flujo Normal: | <ol style="list-style-type: none"> 1. La directora selecciona el colaborador y da clic en 'Modificar Maestro' 2. El sistema despliega 'Nombre', 'Apellidos', 'Fecha de Nacimiento', 'Genero', 'Correo Electrónico', 'Teléfono', 'Teléfono Móvil', 'Domicilio', 'Ciudad', 'Código Postal', 'Estado' y 'Foto' del colaborador 3. La directora selecciona el campo, captura la modificación y da clic en 'Guardar' 4. El sistema actualiza el registro y despliega un mensaje: "El maestro ha sido actualizado" |
| Flujo Alternativo: | <p>FA 1 - Campos Inválidos</p> <ol style="list-style-type: none"> 3.1 En caso de que los campos del registro modificado no han sido valido entonces, 3.2 El sistema despliega un mensaje de advertencia: "El campo no es correcto" <p>FA 2 – Actualizar Foto</p> <ol style="list-style-type: none"> 3.2 En caso de que la directora de clic en 'Actualizar foto' entonces, La directora selecciona la foto desde el explorador de archivos y da clic en 'Establecer foto' |
| Excepción: | <p>EX 1 – Error de conexión</p> <ol style="list-style-type: none"> 4.1 Si el servidor no ha sido conectado entonces, El sistema despliega un mensaje de error: "El servidor no ha sido conectado" |
| Postcondiciones: | POST-1 El registro del maestro ha sido actualizado en la base de datos |

| | |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-06 |
| Nombre: | Dar de baja Maestro |
| Autor: | Jonathan Garcia Delgado |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | El sistema deberá permitir a la directora, inhabilitar el registro del maestro colaborador en caso de baja temporal o definitiva |
| Precondiciones: | PRE-1 El maestro ha sido registrado PRE-2 El sistema ha realizado el CU18 Buscar Maestro PRE-3 El alumno no tiene asociado grupos |
| Flujo Normal: | 1. La directora selecciona el colaborador y da clic en ‘Eliminar Maestro’ 2. El sistema despliega ‘Fecha de ingreso’, ‘Numero de colaborador’, ‘Foto’, ‘Nombre Completo’, ‘Correo Electrónico’ y ‘Teléfonos’ 3. La directora captura el motivo de la baja y da clic en ‘Dar de baja’ 4. El sistema despliega una pantalla de confirmación e inhabilita el registro 5. El sistema despliega un mensaje: “El maestro ha sido dado de baja” |
| Flujo Alterno: | FA 1 – Cancelar Baja 4.1 En caso de que la directora de clic en ‘No’ entonces, El sistema ejecuta el caso de uso CU18 Buscar Maestros |
| Excepción: | EX 1 – Error de conexión 4.1 Si el servidor no ha sido conectado entonces, El sistema despliega un mensaje de error: “El servidor no ha sido conectado” |
| Postcondiciones: | POST-1 El estado del maestro ha sido cambiado a baja POST- 2 El registro del maestro ha sido actualizado en la base de datos |

| | |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-07 |
| Nombre: | Crear Grupo |
| Autor: | Ehecatl Said Cortina Gamboa |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | Este caso de uso inicia cuando la directora quiere crear un nuevo grupo de una determinada clase de baile en algún horario que desee. |
| Precondiciones: | PRE-1.-Debe estar registrado el tipo de baile. PRE-2.-El maestro debe estar registrado. |
| Flujo Normal: | 1.-La directora debe ingresar al menú y seleccionar Grupo. 2.-El sistema despliega un menú de opciones. 3.-La directora elige crear grupo. 4.-El sistema despliega una pantalla con campos para ser llenados, como nombre, horario, quién lo imparte, el salón y el tipo de baile. 5.-La directora ingresa los datos solicitados y selecciona registrar grupo. 6.-El sistema manda un cuadro de confirmación. 7.-La directora selecciona confirmar. 8.-El sistema despliega un mensaje de “Grupo creado exitosamente”. |
| Flujo Alternativo: | FA-1 Cancelar Crear Grupo 5.-La directora ingresa los datos solicitados y selecciona registrar grupo. 6.-El sistema manda un cuadro de confirmación. 7.-La directora selecciona Cancelar. 8.-El sistema despliega un mensaje de “Creación de grupo Cancelado”. Crear Grupo con campos faltantes 5.-La directora ingresa algunos datos solicitados y selecciona registrar grupo. 6.-El sistema manda una advertencia de que algunos campos faltan de ser llenados y una confirmación para continuar. 7.-La directora selecciona confirmar. 8.-El sistema despliega un mensaje de “Grupo creado exitosamente”. |
| Excepción | EX-1 Desconexión con la base de datos. 1.-La directora debe ingresar al menú y seleccionar Grupos. 2.-El sistema despliega un menú de opciones. 3.-La directora elige Crear Grupo. |

| | |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>4.-El sistema despliega una pantalla con campos para ser llenados, como nombre, horario, quién lo imparte, el salón y el tipo de baile.</p> <p>5.-La directora ingresa los datos solicitados y selecciona registrar grupo</p> <p>6.-El sistema despliega un mensaje “la conexión con la base de datos no se pudo establecer y regresa a la directora al menú principal.</p> |
| Postcondiciones | <p>POST1.-Grupo de baile creado.</p> <p>POST2.-Los estudiantes pueden inscribirse a ese grupo.</p> |

| | |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-08 y CU-09 |
| Nombre: | Modificar Grupo |
| Autor: | Ehecatl Said Cortina Gamboa |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | Este caso de uso inicia cuando la directora necesita modificar los datos de un grupo o en su caso eliminarlo por falta de alumnos o porque la clase no se dará un tiempo. |
| Precondiciones: | PRE-1 El grupo debe estar registrado. PRE-2 Algún maestro debe tenerlo a cargo. |
| Flujo Normal: | <p>Modificar Grupo Exitosamente</p> <ol style="list-style-type: none"> 1.-La directora debe ingresar al menú y seleccionar Grupo. 2.-El sistema despliega un menú de opciones. 3.-La directora elige modificar grupo. 4.-El sistema despliega una pantalla de búsqueda para el nombre del grupo. 5.-La directora ingresa los datos el nombre del grupo y selecciona buscar. 6.-El sistema muestra una pantalla donde se muestran los grupos que coincidan con esas características. 7.-La directora selecciona un grupo y elige modificar. 8.-El sistema muestra una pantalla donde se muestran los datos del grupo donde se podrán modificar. 9.-La directora ingresa los nuevos datos y selecciona guardar. 10.-El sistema solicita una confirmación para continuar. 11.-La directora elige confirmar y guardar. 12.-El sistema despliega un mensaje de “Datos del grupo guardado exitosamente”. <p>Eliminar Grupo Exitosamente</p> <ol style="list-style-type: none"> 1.-La directora debe ingresar al menú y seleccionar Grupo. 2.-El sistema despliega un menú de opciones. 3.-La directora elige eliminar grupo. 4.-El sistema despliega una pantalla de búsqueda para el nombre del grupo. 5.-La directora ingresa los datos el nombre del grupo y selecciona buscar. 6.-El sistema muestra una pantalla donde se muestran los grupos que coincidan con esas características. 7.-La directora elige un grupo selecciona eliminar. |

| | |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>8.-El sistema solicita la confirmación para eliminar.</p> <p>9.-La directora elige confirmar.</p> <p>8.-El sistema despliega un mensaje de “Grupo eliminado exitosamente”.</p> |
| Flujo Alternativo: | <p>FA-1 Cancelar Modificar Grupo</p> <p>7.-La directora selecciona un grupo y elige modificar.</p> <p>8.-El sistema muestra una pantalla donde se muestran los datos del grupo donde se podrán modificar.</p> <p>9.-La directora ingresa los nuevos datos y selecciona guardar.</p> <p>10.-El sistema solicita una confirmación para continuar.</p> <p>11.-La directora elige cancelar.</p> <p>12.-El sistema regresa al menú principal.</p> <p>FA-2 No hay grupo para Modificar</p> <p>5.-La directora ingresa los datos el nombre del grupo y selecciona buscar.</p> <p>6.-El sistema despliega un mensaje de “no existe el grupo”.</p> <p>Fallo al Modificar Grupo</p> <p>7.-El sistema muestra una pantalla donde se muestran los datos del grupo donde se podrán modificar.</p> <p>8.-La directora ingresa los nuevos datos y selecciona guardar.</p> <p>9.-El sistema despliega un mensaje de “hay datos incorrectos que no se pueden guardar.”.</p> <p>Cancelar Eliminar Grupo</p> <p>9.-La directora elige Cancelar.</p> <p>8.-El sistema regresa al menú principal.</p> <p>Fallo al eliminar grupo</p> <p>5.-La directora ingresa los datos el nombre del grupo y selecciona buscar.</p> <p>6.- El sistema despliega un mensaje de “El grupo no existe o ya fue eliminado”.</p> |
| Excepción | <p>Desconexión con la base de datos.</p> <p>1.-La directora debe ingresar al menú y seleccionar Grupos.</p> <p>2.-El sistema despliega un menú de opciones.</p> <p>3.-La directora elige Modificar grupo.</p> <p>4.-El sistema despliega una pantalla de búsqueda.</p> <p>5.-La directora ingresa los datos solicitados y selecciona buscar grupo.</p> <p>6.-El sistema despliega un mensaje “la conexión con la base de datos no se pudo establecer y regresa a la directora al menú principal.</p> |
| Postcondiciones | <p>POST-1. Los datos del grupo solicitado han sido modificados.</p> <p>POST-2. El grupo seleccionado pudo ser eliminado.</p> |

| | |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-10 |
| Nombre: | Pagar Maestro |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La directora registrará el pago de un maestro |
| Precondiciones: | PRE-1 Existe maestro en el sistema PRE-2 Maestro tiene asignados grupo(s) |
| Flujo Normal: | <ol style="list-style-type: none"> 1. Empieza cuando la directora visita la pestaña de “Maestro” y selecciona “Pagar a Maestro” 2. Sistema despliega campo de búsqueda de maestro 3. Directora ingresa dato de búsqueda del maestro y da clic en buscar 4. Sistema despliega coincidencias 5. Directora selecciona maestro y da clic en pagar 6. Sistema calcula el monto de pago y muestra sugerencia de pago 7. Directora da clic en pagar 8. Sistema registra el pago en el sistema 9. Directora logra realizar el pago |
| Flujo Alternativo: | <p>FA 1 – No se encuentra maestro</p> <ol style="list-style-type: none"> 4.1 Sistema despliega mensaje “No existe coincidencias” 4.2 Directora continua en el Flujo normal 3 <p>FA 2 – Cambiar monto a pagar</p> <ol style="list-style-type: none"> 7.1 Directora edita el campo de pago y da clic en pagar 7.2 Sistema registra el pago en el sistema 7.3 Directora logra realizar el pago |
| Excepción | <p>EX 1 – No existe conexión con la Base de Datos</p> <ol style="list-style-type: none"> 7.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde” 7.2 Directora continua en el Flujo normal 3 |
| Postcondiciones | POST-1 Se registra pago en la base de datos |

| | |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-11 |
| Nombre: | Registrar Egreso |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La Directora desea registrar un egreso en el sistema |
| Precondiciones: | Ninguno |
| Flujo Normal: | <ol style="list-style-type: none"> 1. Empieza cuando la Directora visita la pestaña de “Egresos e Ingresos” y selecciona “Registrar egreso” 2. Sistema despliega campos para ingresar monto y motivo 3. Directora ingresa los datos y da clic en “Registrar” 4. Sistema registra el egreso y muestra mensaje de “éxito” 5. Directora logra realizar un registro de egreso |
| Flujo Alterno: | <p>FA 1 - No se pudo registrar por campos inválidos</p> <ol style="list-style-type: none"> 4.1 Sistema no puede guardar el registro y muestra mensaje de “Error” 4.2 Usuario acepta el mensaje 4.3 Sistema continua en el Flujo normal 3 |
| Excepción | <p>EX 1 – No existe conexión con la base de datos</p> <ol style="list-style-type: none"> 4.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde” 4.2 Directora continua en el Flujo normal 3 |
| Postcondiciones | POST-1 Se crea registro de egreso |

| | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-12 |
| Nombre: | Ver historial de egresos |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La Directora desea ver todos los egresos |
| Precondiciones: | Ninguno |
| Flujo Normal: | 1. Empieza cuando Directora visita la pestaña de “Egresos e Ingresos” y da clic en “Historial de Egresos” 2. Sistema despliega historial de Egresos en forma de tabla y el total acumulado 3. Directora logra ver historial de egresos |
| Flujo Alternativo: | FA-1 Ver historial de Egresos de un mes en específico 3.1 Directora selecciona un mes 3.2 Sistema muestra los egresos de ese mes y el total acumulado 3.3 Directora logra ver historial de egresos FA-2 Ver historial de egresos de un año en específico 3.1 Directora selecciona un año 3.2 Sistema muestra los egresos de ese año y el total acumulado 3.3 Directora logra ver historial de egresos |
| Excepción | EX 1 – No existe conexión con la base de datos 4.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde” |
| Postcondiciones | Ninguna |

| | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-13 |
| Nombre: | Ver historial de ingresos |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La Directora desea ver todos los ingresos |
| Precondiciones: | Ninguno |
| Flujo Normal: | 1. Empieza cuando Directora visita la pestaña de “Egresos e Ingresos” y da clic en “Historial de Ingresos” 2. Sistema despliega historial de Ingresos en forma de tabla y el total acumulado 3. Directora logra ver historial de Ingresos |
| Flujo Alternativo: | FA-1 Ver historial de Ingresos de un mes en específico 3.1 Directora selecciona un mes 3.2 Sistema muestra los ingresos de ese mes y el total acumulado 3.3 Directora logra ver historial de ingresos FA-2 Ver historial de ingresos de un año en específico 3.1 Directora selecciona un año 3.2 Sistema muestra los ingresos de ese año y el total acumulado 3.3 Directora logra ver historial de ingresos |
| Excepción | EX 1 – No existe conexión con la base de datos 4.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde” |
| Postcondiciones | Ninguna |

| | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-14 |
| Nombre: | Ver Notificaciones |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La Directora desea ver notificaciones de pago de alumnos o avisos de próximas clases |
| Precondiciones: | PRE-1 Existen Alumnos PRE-2 Existen Grupos |
| Flujo Normal: | 1. Empieza cuando la directora visita la pestaña de “Otros y da clic en ”notificaciones 2. Sistema muestra notificaciones de pago y clases 3. Directora logra ver las notificaciones |
| Flujo Alternativo: | Ninguno |
| Excepción | EX 1 – No existe conexión con la base de datos 4.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde” |
| Postcondiciones | Ninguna |

| | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-15 |
| Nombre: | Crear Promoción |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La Directora desea crear una promoción |
| Precondiciones: | Ninguna |
| Flujo Normal: | <ol style="list-style-type: none"> 1. Empieza cuando la Directora visita la pestaña de “Otros” y da clic en “Promociones” 2. Sistema despliega campos de tipo, porcentaje de descuento y descripción 3. Directora da clic en nuevo y llena los campos y da clic en “Registrar” 4. Sistema guarda la promoción en el sistema y notifica éxito al registrar 5. Directora logra crear una promoción |
| Flujo Alternativo: | <p>EX 1 -Descripción vacía</p> <ol style="list-style-type: none"> 4.1 Sistema no puede guardar la promoción, sistema notifica 4.2 Directora acepta la notificación 4.3 Sistema continua en el paso 3 del flujo normal |
| Excepción | <p>EX 1 – No existe conexión con la base de datos</p> <ol style="list-style-type: none"> 4.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde” |
| Postcondiciones | POST-1 Se crea una promoción en el sistema |

| | |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-16 |
| Nombre: | Modificar Promoción |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La Directora desea modificar una promoción |
| Precondiciones: | PRE-1 Existan promociones |
| Flujo Normal: | <ol style="list-style-type: none"> 1. Empieza cuando la Directora visita la pestaña de “Otros” y da clic en “Promociones” 2. Sistema despliega las promociones existentes 3. Directora selecciona una promoción 4. Sistema despliega la información de la promoción en los campos editables. 5. Directora llena los campos y da clic en “Actualizar” 6. Sistema guarda los cambios de la promoción en el sistema y notifica éxito al actualizar 7. Directora logra actualizar una promoción |
| Flujo Alternativo: | <p>EX 1 -Descripción vacía</p> <ol style="list-style-type: none"> 6.1 Sistema no puede guardar la promoción, sistema notifica 6.2 Directora acepta la notificación 6.3 Sistema continua en el paso 5 del flujo normal |
| Excepción | <p>EX 1 – No existe conexión con la base de datos</p> <ol style="list-style-type: none"> 4.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde” |
| Postcondiciones | POST-1 Se modifica una promoción en el sistema |

| | |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-17 |
| Nombre: | Eliminar Promoción |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La Directora desea eliminar una promoción |
| Precondiciones: | PRE-1 Existan Promociones |
| Flujo Normal: | <ol style="list-style-type: none"> 1. Empieza cuando la Directora visita la pestaña de “Otros” y da clic en “Promociones” 2. Sistema despliega las promociones existentes 3. Directora selecciona una promoción 4. Sistema despliega la información de la promoción en los campos editables. 5. Directora da clic en “Eliminar” 6. Sistema elimina la promoción del sistema y notifica éxito al eliminar 7. Directora logra eliminar una promoción |
| Flujo Alternativo: | Ninguno |
| Excepción | EX 1 – No existe conexión con la base de datos 4.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde” |
| Postcondiciones | POST-1 Se elimina una promoción del sistema |

| | |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-18 |
| Nombre: | Registrar Asistencia |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La directora desea registrar la asistencia de los alumnos de un grupo |
| Precondiciones: | PRE-1 Existan Grupos |
| Flujo Normal: | <ol style="list-style-type: none"> 1. Empieza cuando la Directora visita la pestaña de “Grupo” y da clic en “Pase de Asistencia” 2. Sistema despliega los grupos 3. Directora selecciona el grupo 4. Sistema despliega alumnos de ese grupo 5. Directora marca alumnos que asistieron y da clic en guardar 6. Sistema guarda la lista de asistencia 7. Directora logra registrar asistencia |
| Flujo Alternativo: | Ninguno |
| Excepción | <p>EX 1 – No existe conexión con la base de datos</p> <p>6.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde”</p> |
| Postcondiciones | POST-1 Se registra lista de asistencia |

| | |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU-19 |
| Nombre: | Cambiar alumno de grupo |
| Autor: | Mauricio Juárez Capistran |
| Fecha de Creación: | 14/02/17 |
| Actores: | Directora |
| Descripción: | La directora desea cambiar a un alumno de grupo |
| Precondiciones: | PRE-1 Existan Grupo y Alumnos |
| Flujo Normal: | <ol style="list-style-type: none"> 1. Empieza cuando la Directora visita la pestaña de “Grupo” y da clic en “Cambiar Alumno” 2. Sistema despliega los grupos origen y grupos destino 3. Directora selecciona el grupo origen 4. Sistema despliega alumnos de ese grupo 5. Directora selecciona el o los alumnos a cambiar 6. Sistema los mueve a una nueva lista 7. Directora selecciona el grupo destino y da clic en “Cambiar Alumno(s)” 8. Sistema cambia al o los alumnos de grupo y notifica éxito al cambiar. 9. Directora logra cambiar a un alumno o alumnos |
| Flujo Alternativo: | Ninguno |
| Excepción | <p>EX 1 – No existe conexión con la base de datos</p> <p>6.1 Sistema despliega mensaje “No hay conexión, inténtalo más tarde”</p> |
| Postcondiciones | POST-1 Se cambia a alumno(s) de grupo |

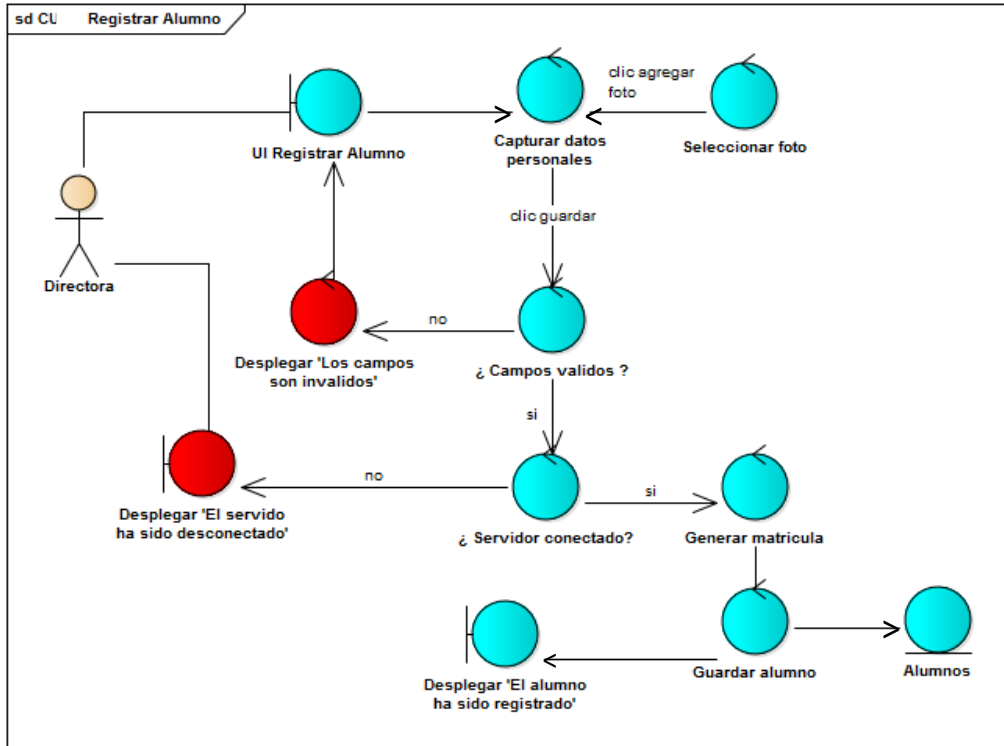
| | |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU 20 |
| Nombre: | Pagar Inscripción |
| Autor: | Ehecatl Said Cortina Gamboa |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | Este caso de uso inicia cuando la directora registra el pago de inscripción de un nuevo estudiante, o en su defecto la reinscripción de algún estudiante ya inscrito. |
| Precondiciones: | PRE-1 El estudiante ya debe estar inscrito. PRE-2 El grupo de danza debe estar registrado. |
| Flujo Normal: | <p>Pagar Inscripción Exitosa</p> <p>1a.-La directora ingresa al menú de pagos. 2a.-El sistema despliega una pantalla de búsqueda. 3a.-La directora ingresa los datos del alumno y seleccionar buscar 4a.-El sistema despliega los resultados de los alumnos que cumplen con el criterio de búsqueda. 5a.-La directora da clic en inscribir. 6a.-El sistema despliega una pantalla nueva donde se ven reflejados, el tipo de clases, los datos de esta. 7a.-La directora selecciona un tipo de danza, y da en buscar. 8a.-El sistema despliega los grupos de danza que concuerdan con el tipo de danza. 9a.-La directora selecciona de los grupos disponibles, el maestro, si es el grupo que se desea da clic en generar pago. 10a.-El sistema despliega una nueva pantalla donde se ven los detalles del pago, las promociones. 11a.-La directora llena los datos según le convenga y da clic en pagar. 12a.-El sistema pide la confirmación del pago. 13a.-La directora da clic en confirmar pago, 14a.-El sistema regresa el mensaje de “El alumno ha pagado su inscripción”, además de que el estado del alumno en pagos pasa a estar al corriente.</p> <p>Pagar Reinscripción Exitosa</p> <p>8b.-El sistema despliega los grupos de danza que concuerdan con el tipo de danza, así mismo los datos de cuando fue el último pago del alumno, pues se hace cada año. 9b.-La directora selecciona de los grupos disponibles, el maestro, si es el grupo que se desea da clic en generar pago. 10b.-El sistema despliega una nueva pantalla donde se ven los detalles del pago, las promociones. 11b.-La directora llena los datos según le convenga y da clic en pagar. 12b.-El sistema pide la confirmación del pago.</p> |

| | |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>13b.-La directora da clic en confirmar pago,</p> <p>14b.-El sistema regresa el mensaje de “El alumno ha pagado su inscripción”, además de que el estado del alumno en pagos pasa a estar al corriente.</p> |
| Flujo Alternativo: | <p>Cancelar Pagar Reinscripción</p> <p>10d.-El sistema despliega una nueva pantalla donde se ven los detalles del pago, las promociones.</p> <p>11d.-La directora llena los datos según le convenga y da clic en pagar.</p> <p>12d.-El sistema pide la confirmación del pago.</p> <p>13d.-La directora da clic en cancelar,</p> <p>14d.-El sistema regresa al menú inicial.</p> |
| Excepción | <p>Desconexión de la Bd.</p> <p>8e.-El sistema despliega un mensaje de que “la conexión a la bd se interrumpió o que no hay conexión”.</p> <p>9e.-La directora da clic en aceptar</p> |
| Postcondiciones | <p>Post1.-El estado del alumno es disponible para recibir clases o apuntarse en algún grupo.</p> <p>Post2.-El registro de pagos del alumno ha sido actualizado.</p> |

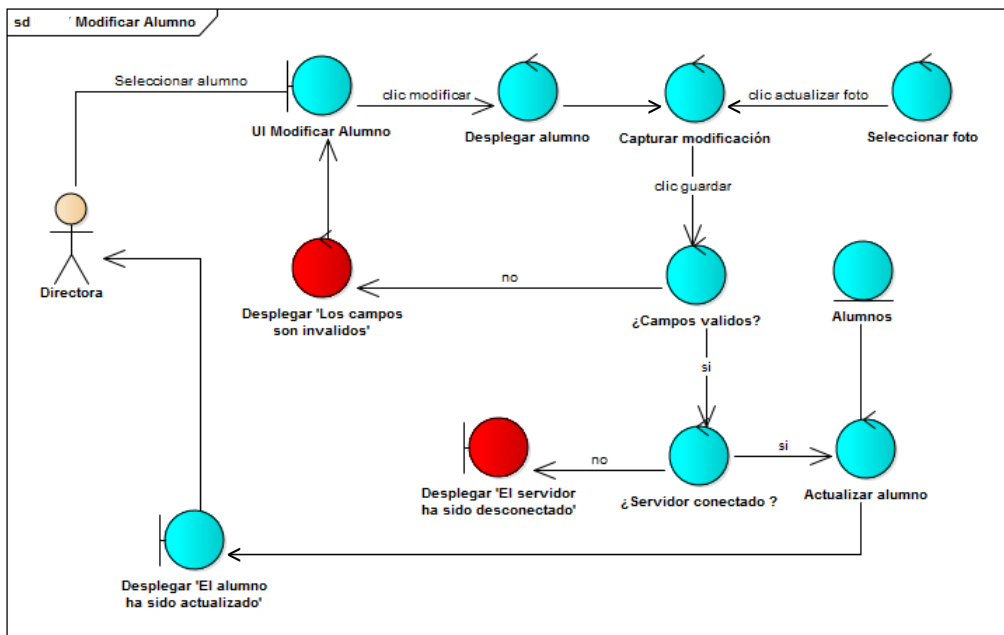
| | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ID: | CU 21 |
| Nombre: | Pagar Mensualidad |
| Autor: | Ehecatl Said Cortina Gamboa |
| Fecha de Creación: | 16/02/17 |
| Actores: | Directora |
| Descripción: | Este Caso de Uso inicia cuando un alumno al iniciar su mes en y el día en el cual se inscribió debe pagar el monto total de su mensualidad por la inscripción de sus clases. |
| Precondiciones: | Pre1.-El alumno debe estar registrado. Pre2.-El alumno debe estar inscrito a alguna clase. |
| Flujo Normal: | Pagar Mensualidad Exitosa 1a.-La directora ingresa al menú de pagos. 2a.-El sistema despliega la ventana de pagos de mensualidad. 3a.-La directora selecciona al alumno del cual pagara su mensualidad. 4a.-El sistema despliega los datos de los grupos a los cuales asiste el alumno. 5a.-La directora selecciona el o las clases a las cuales asiste el alumno y puede o no seleccionar las promociones válidas y da clic en pagar mensualidad. 6a.-El sistema despliega una ventana de confirmación. 7a.-La directora elige confirmar pago de mensualidad. 8a.-El sistema despliega el mensaje de “pago de mensualidad exitoso”. |
| Flujo Alternativo: | Cancelar Pagar Mensualidad 7b.-La directora elige cancelar pago de mensualidad. 8b.-El sistema despliega el mensaje de “pago de mensualidad cancelado”. |
| Excepción | Desconexión de la Bd. 2c.- El sistema despliega un mensaje “la conexión a la base de datos no es posible”. 3c.-La directora da clic en aceptar. 4c.- El sistema regresa al menú principal. |
| Postcondiciones | Post1.-El alumno paga la mensualidad Post2.-El historial de pagos del alumno esta actualizado. |

Diagramas de Robustez

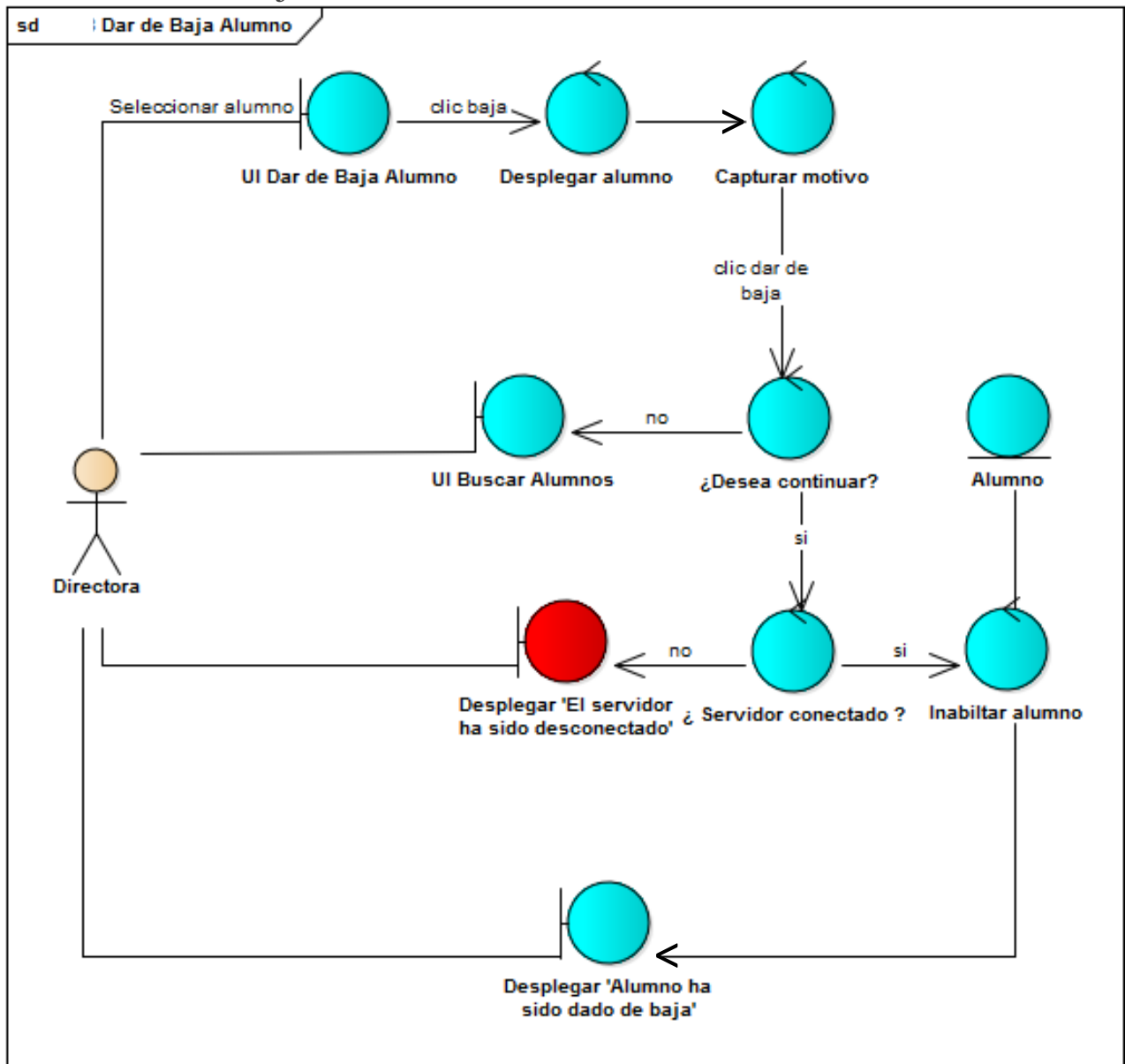
CU-01. Registrar Alumno



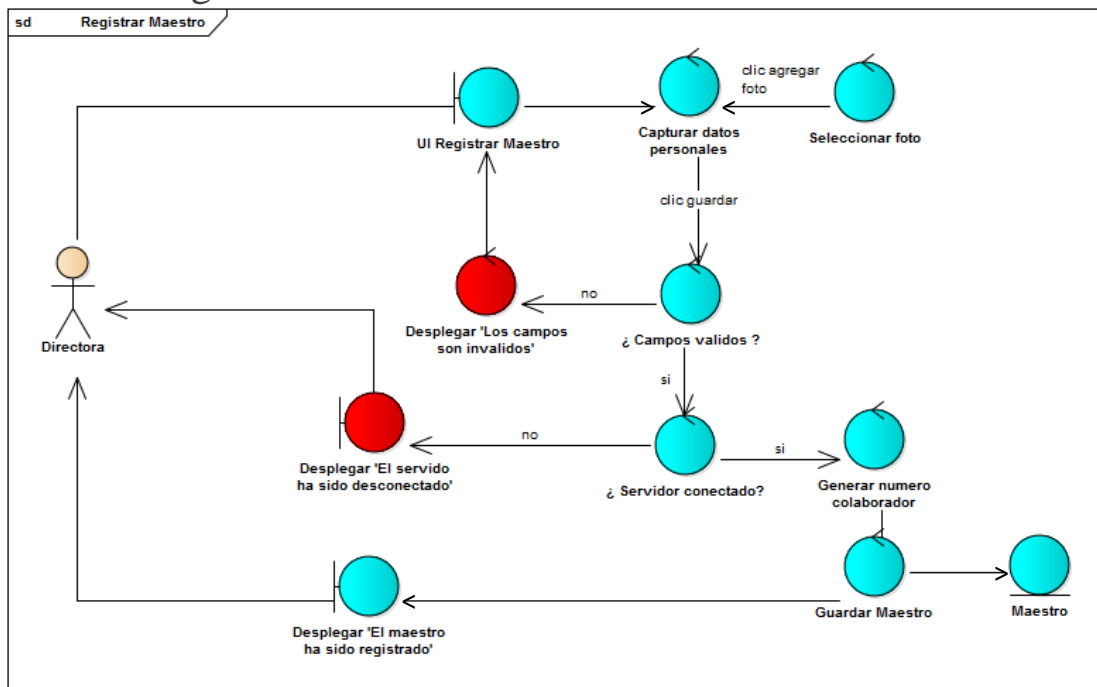
CU-02. Modificar Alumno



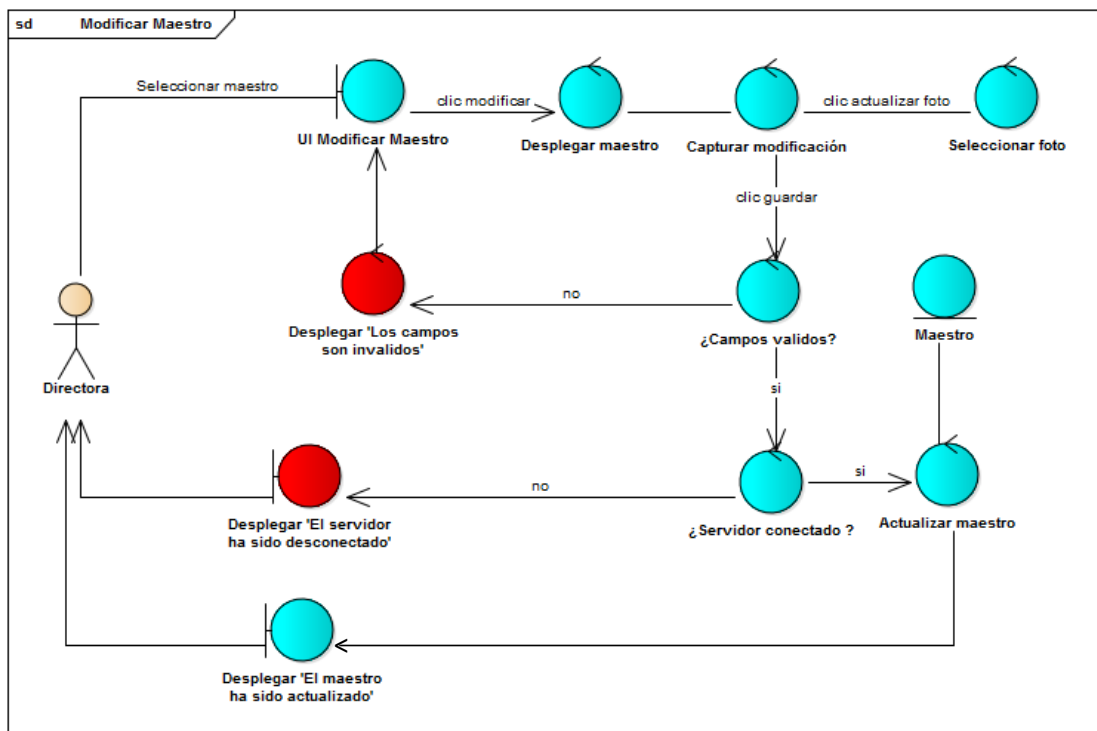
CU-03. Dar de Baja Alumno



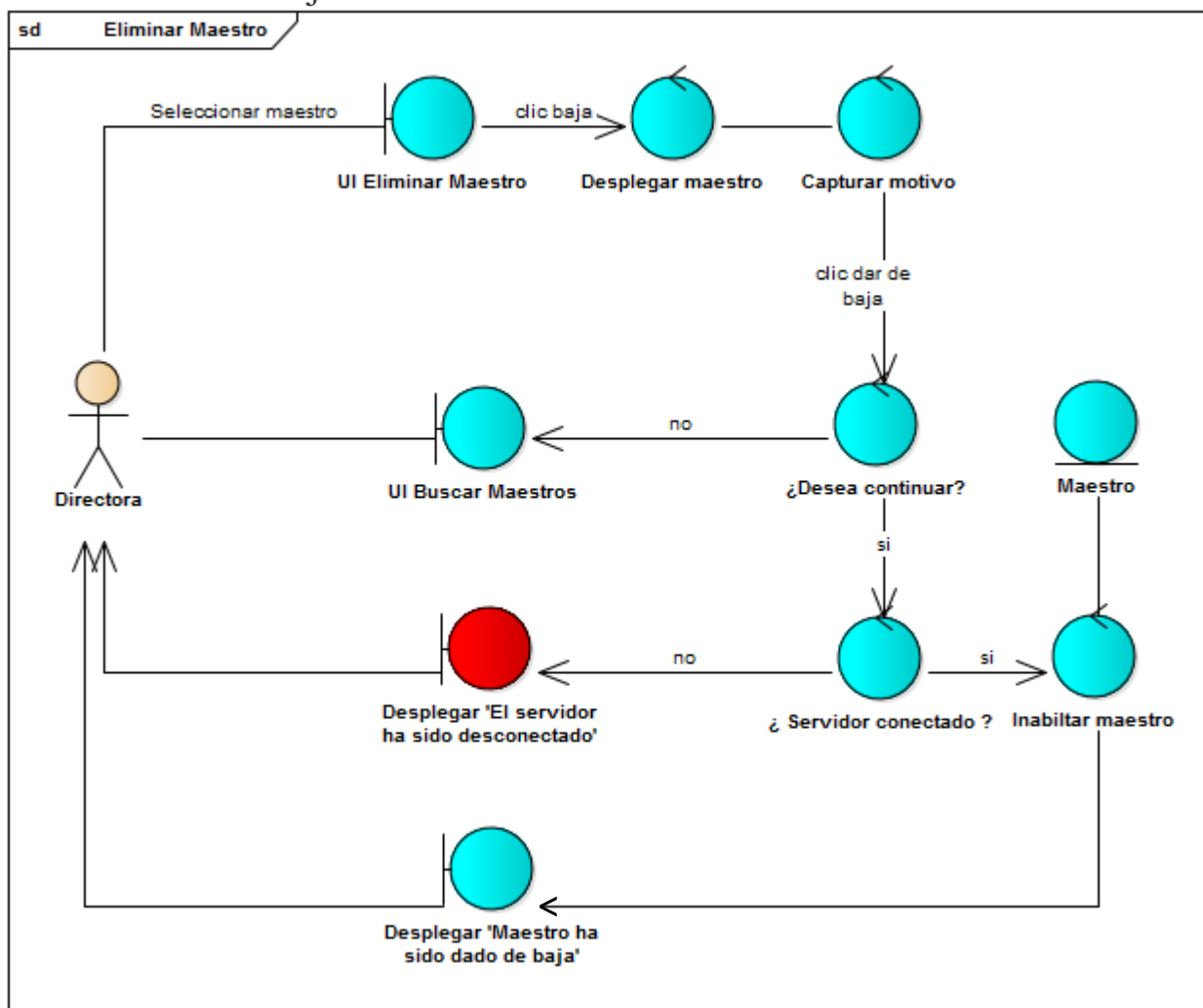
CU-04. Registrar Maestro



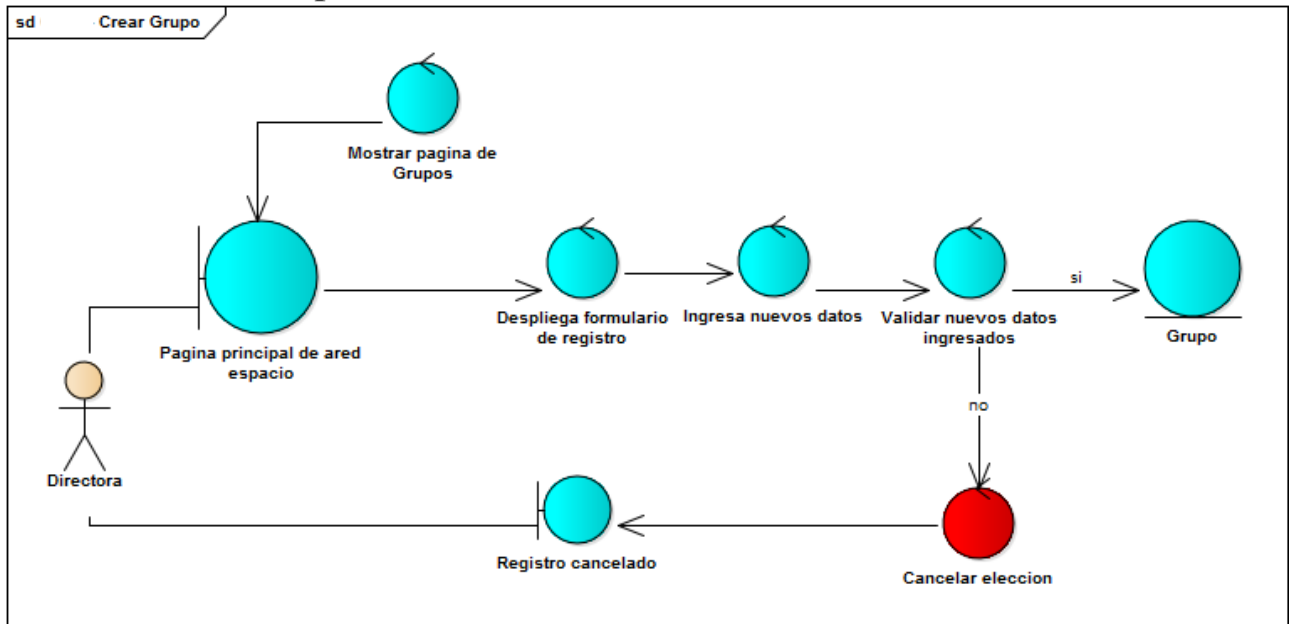
CU-05. Modificar Maestro



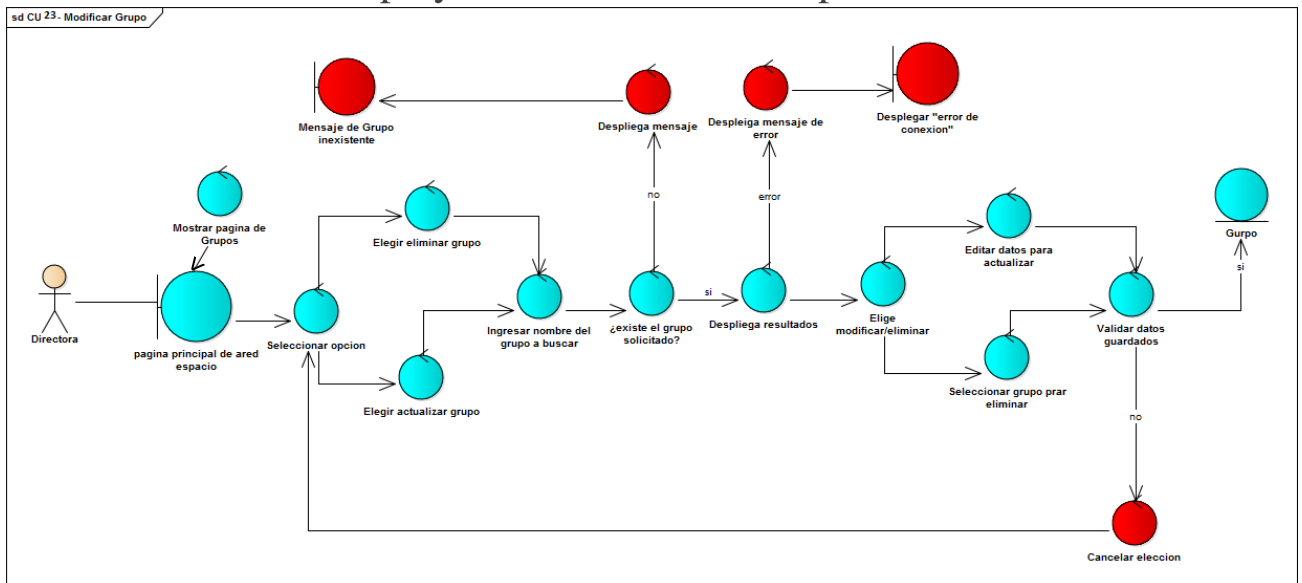
CU-06. Dar de Baja Maestro



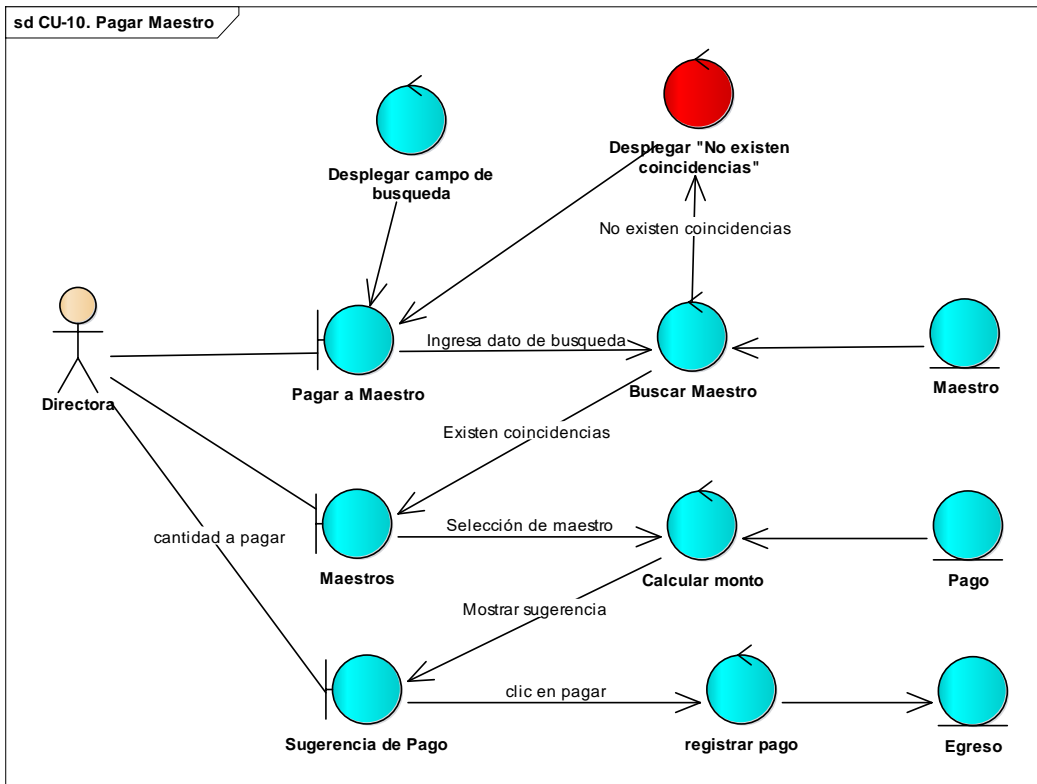
CU-07. Crear Grupo



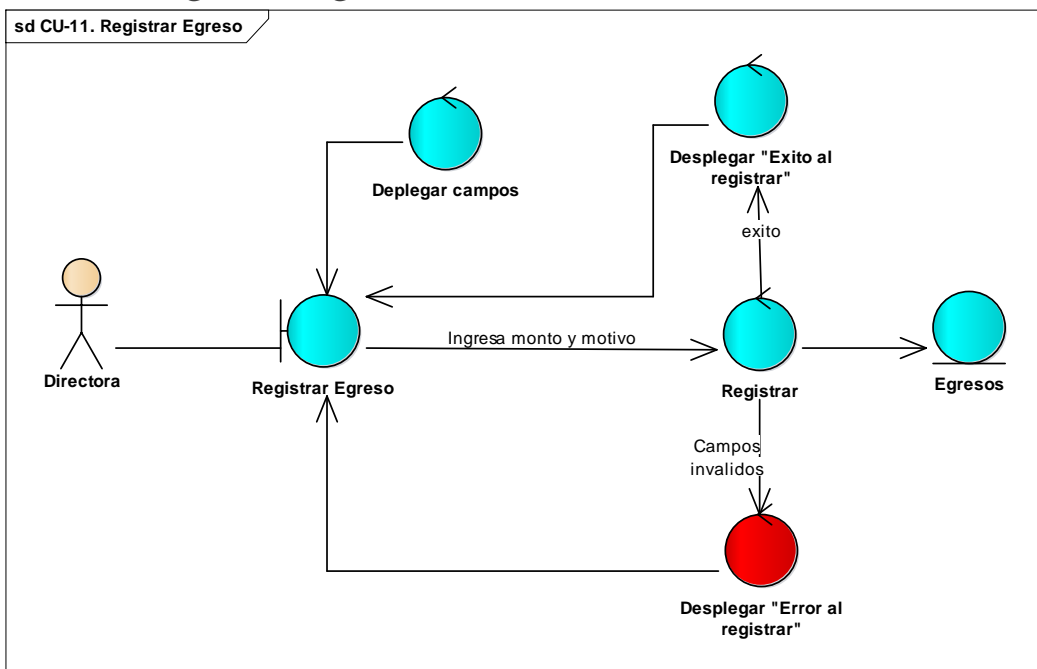
CU-08 Modificar Grupo y CU-09 Eliminar Grupo



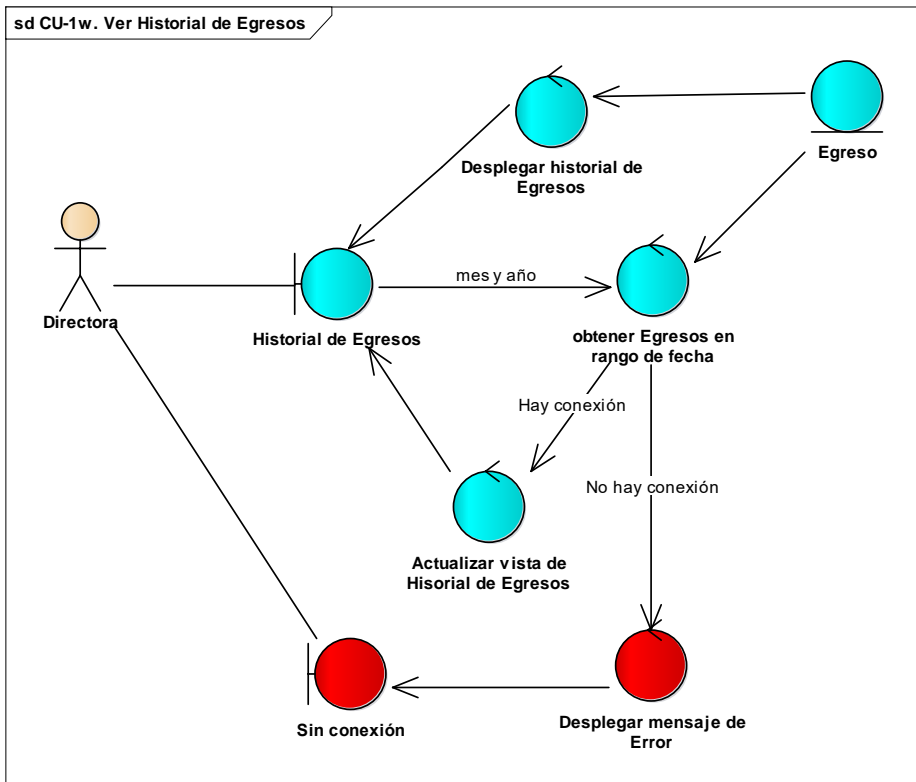
CU-10. Pagar Maestro



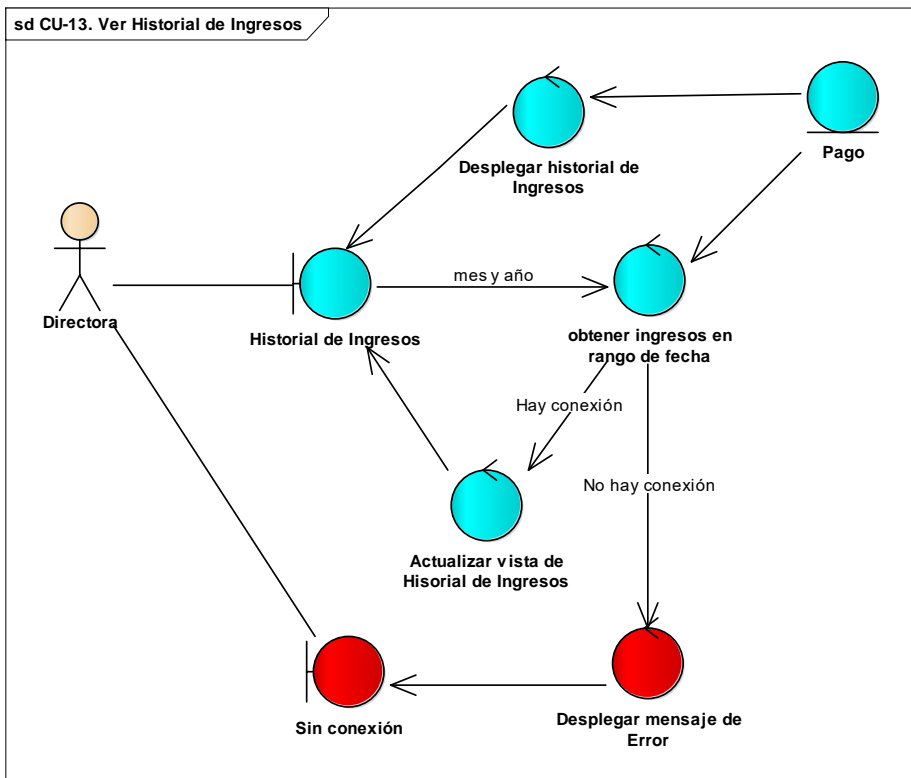
CU-11. Registrar Egreso



CU-12. Ver historial de Egresos

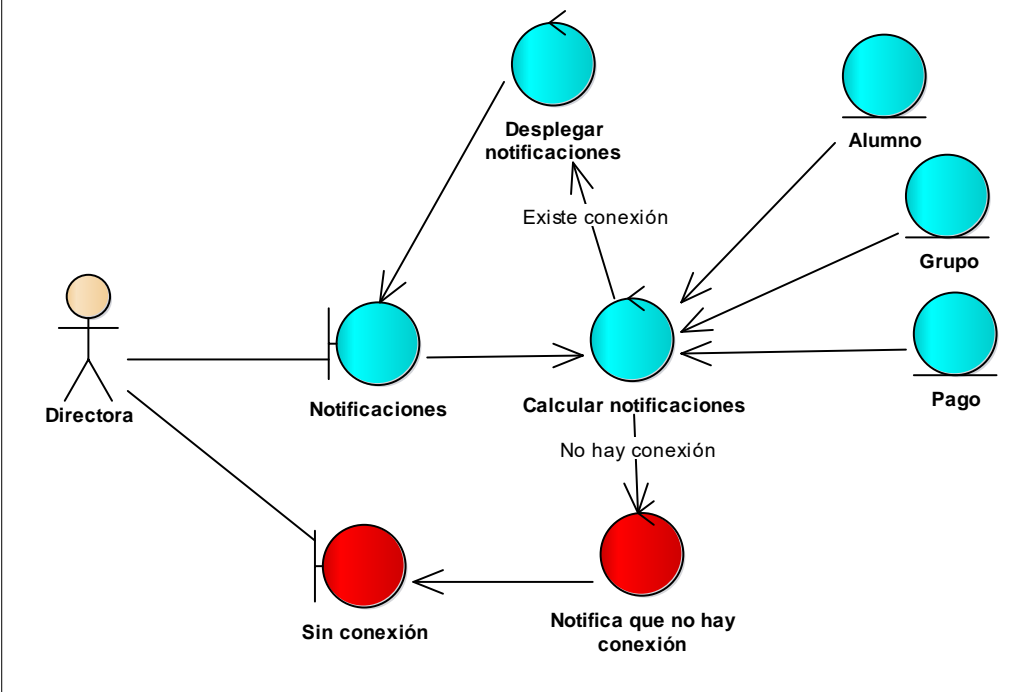


CU-13. Ver historial de Ingresos



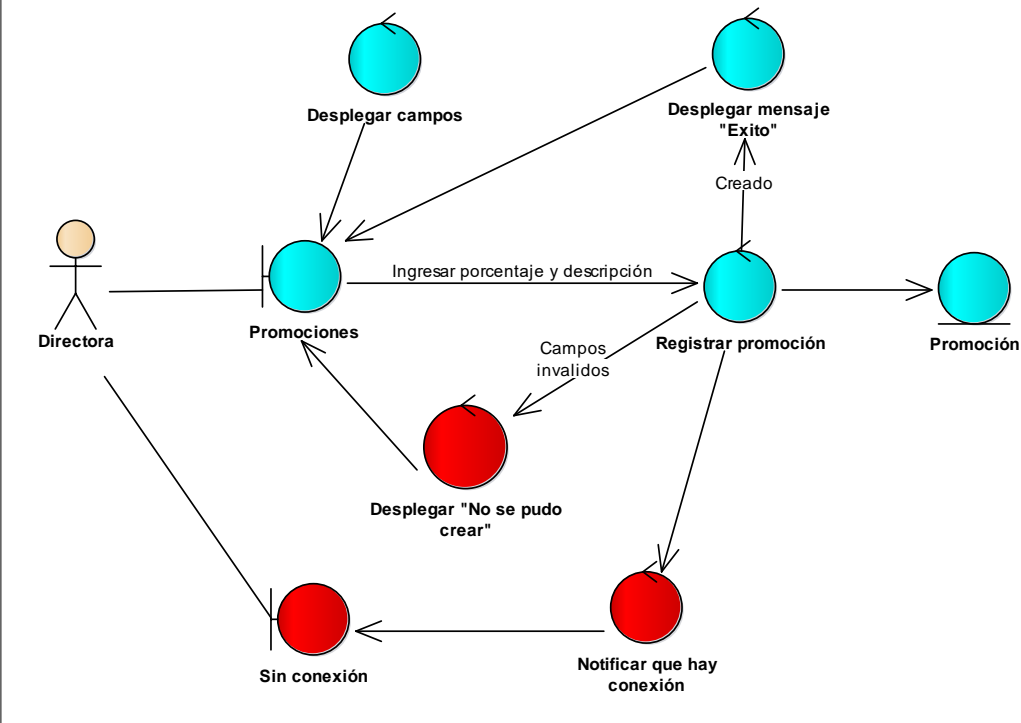
CU-14. Ver Notificaciones

sd CU-14. Ver Notificaciones

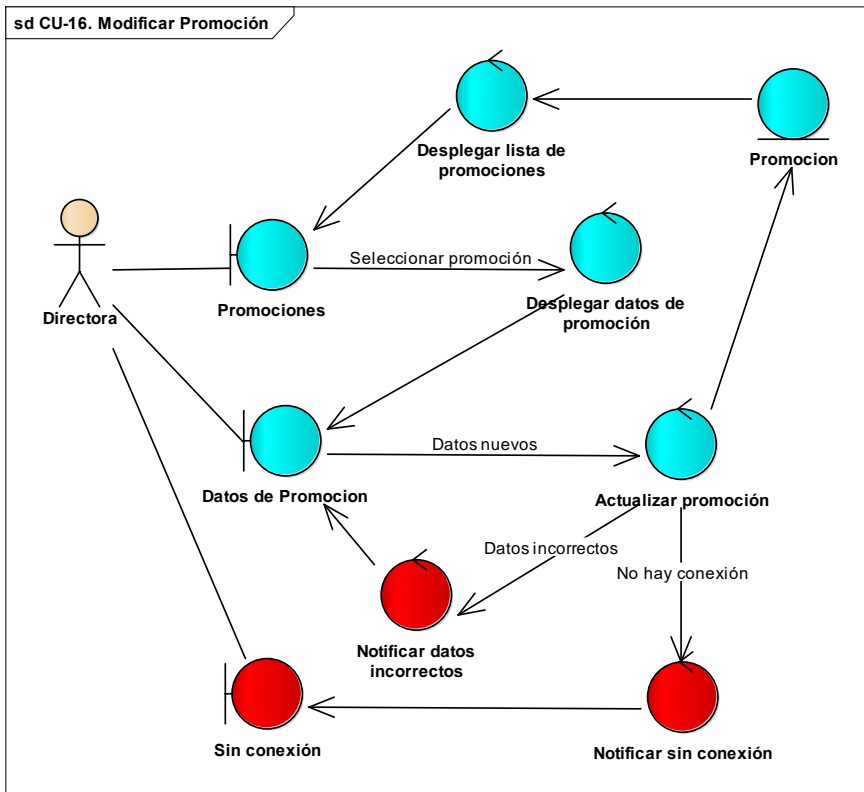


CU-15. Crear Promoción

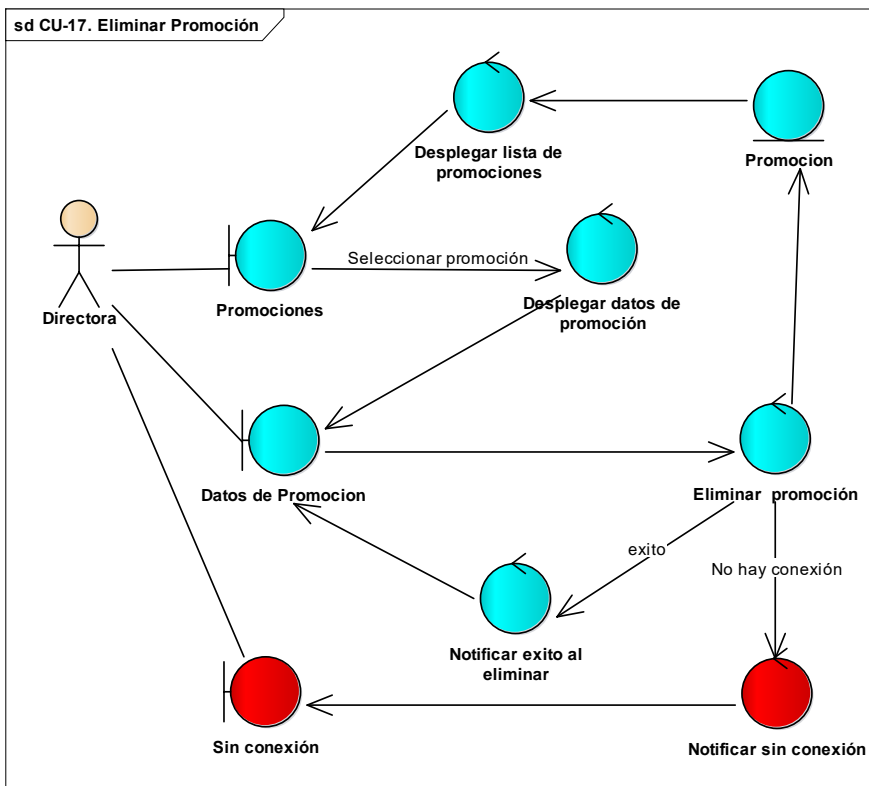
sd CU-15. Crear Promoción



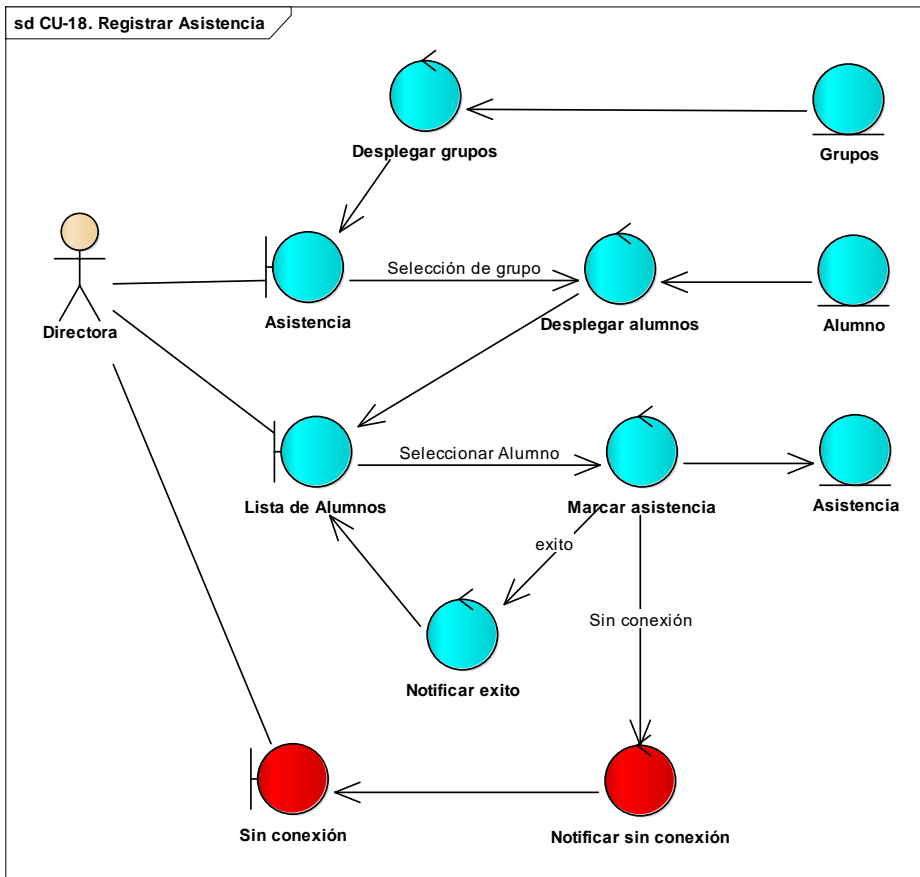
CU-16. Modificar Promoción



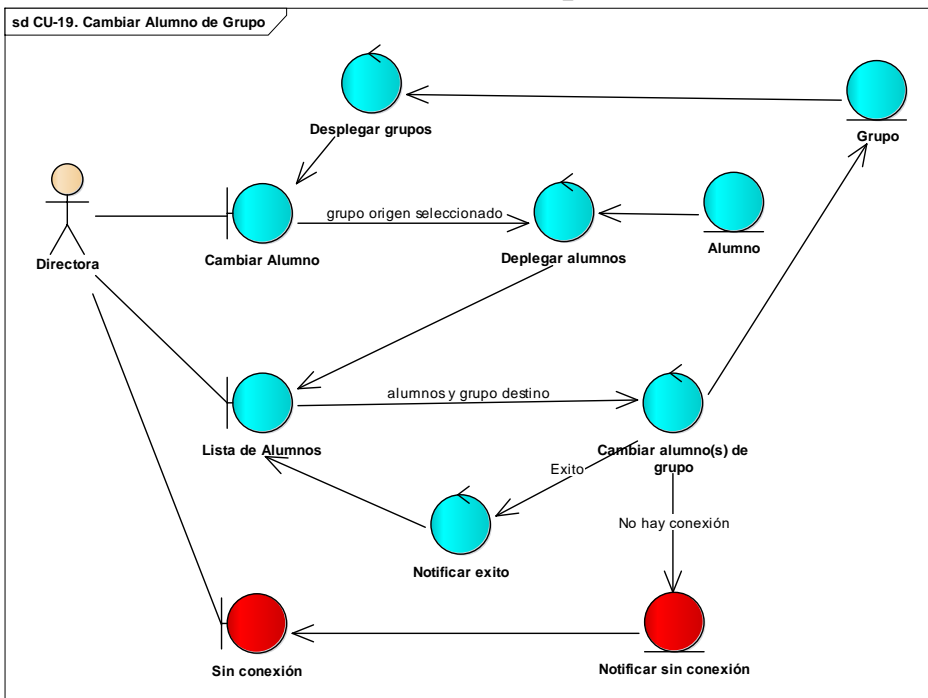
CU-17. Eliminar Promoción



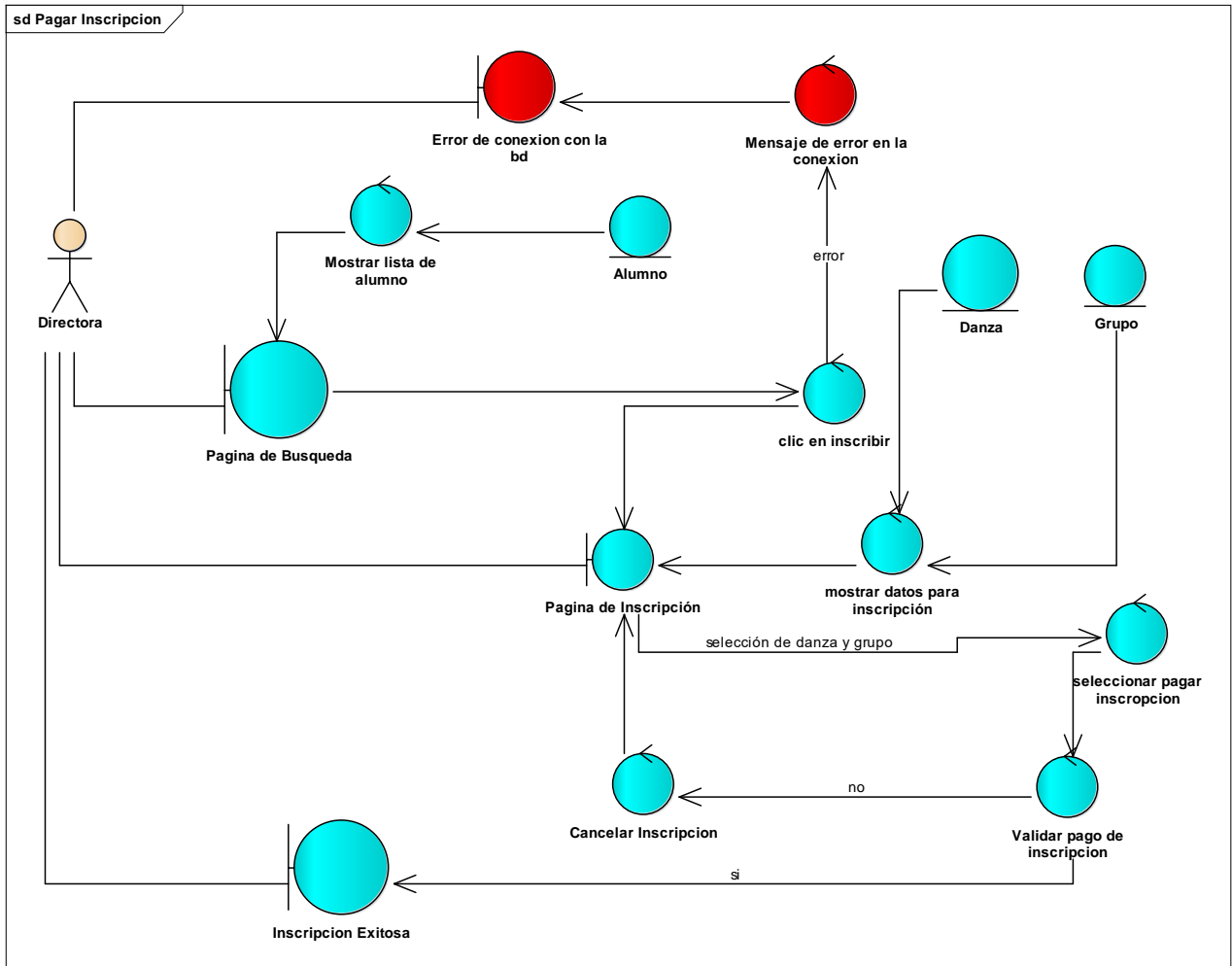
CU-18. Registrar Asistencia



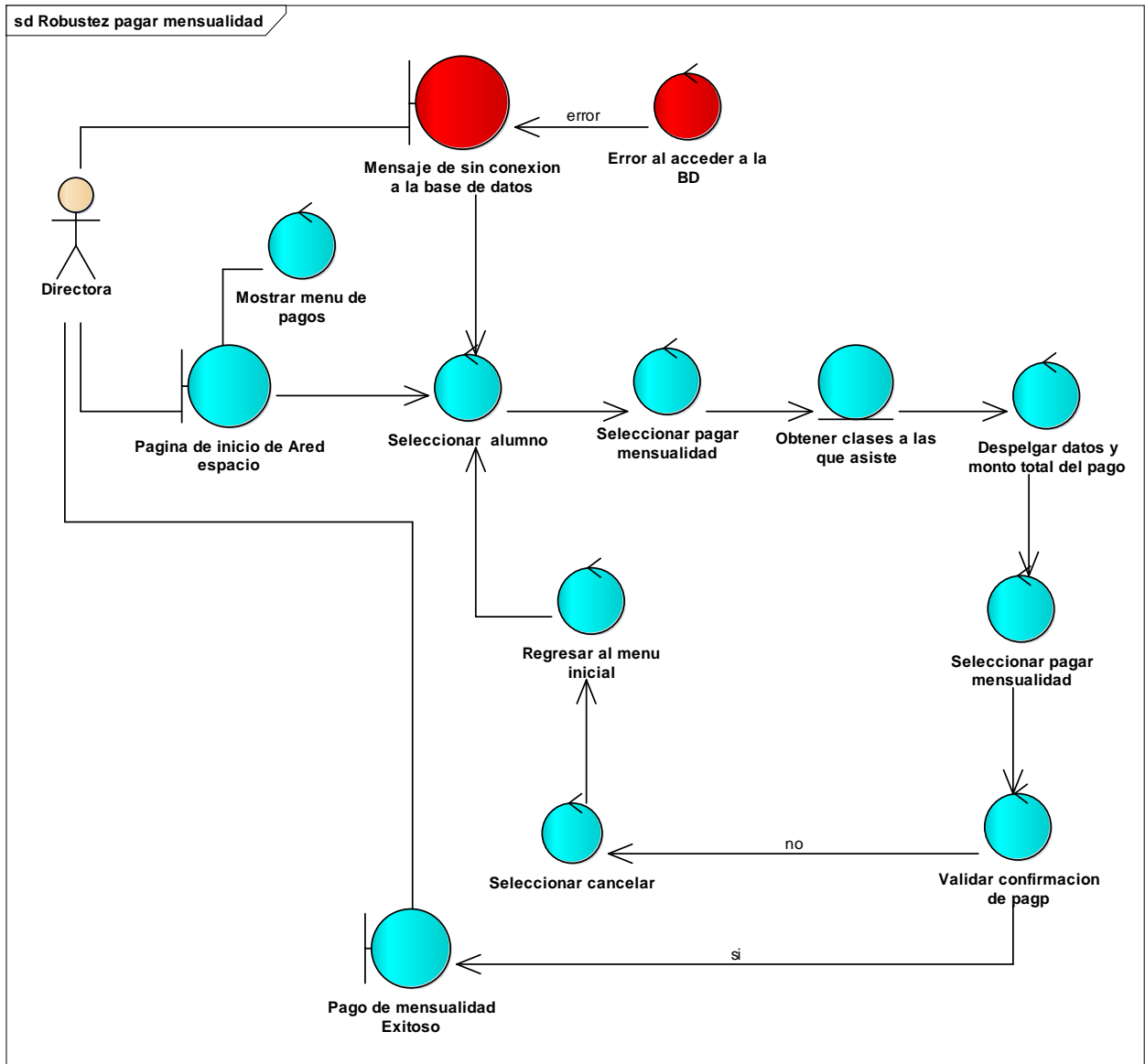
CU-19. Cambiar Alumno de Grupo



CU-20 Inscribir Alumno



CU-21 Pagar Mensualidad



Modelo de Dominio

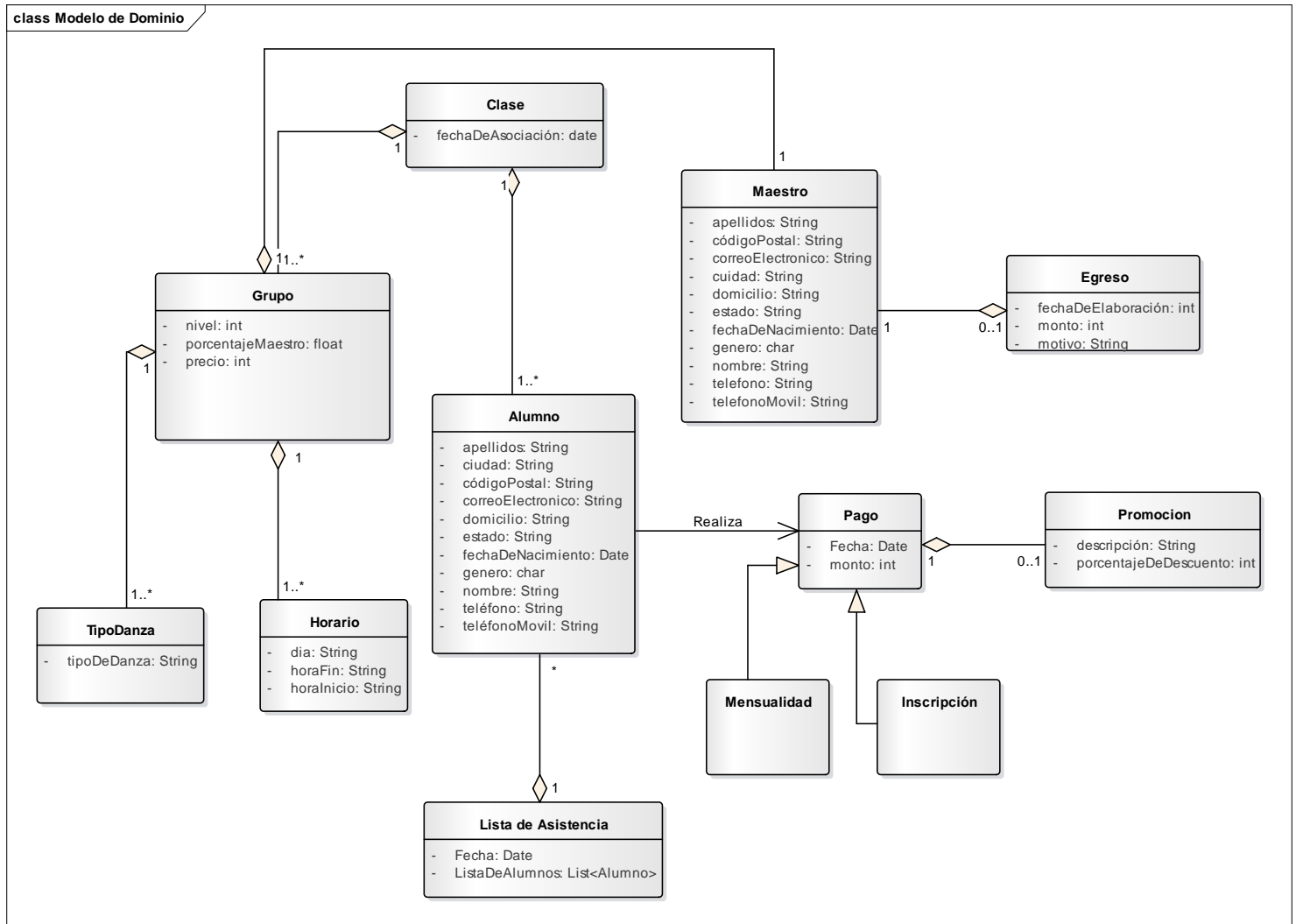
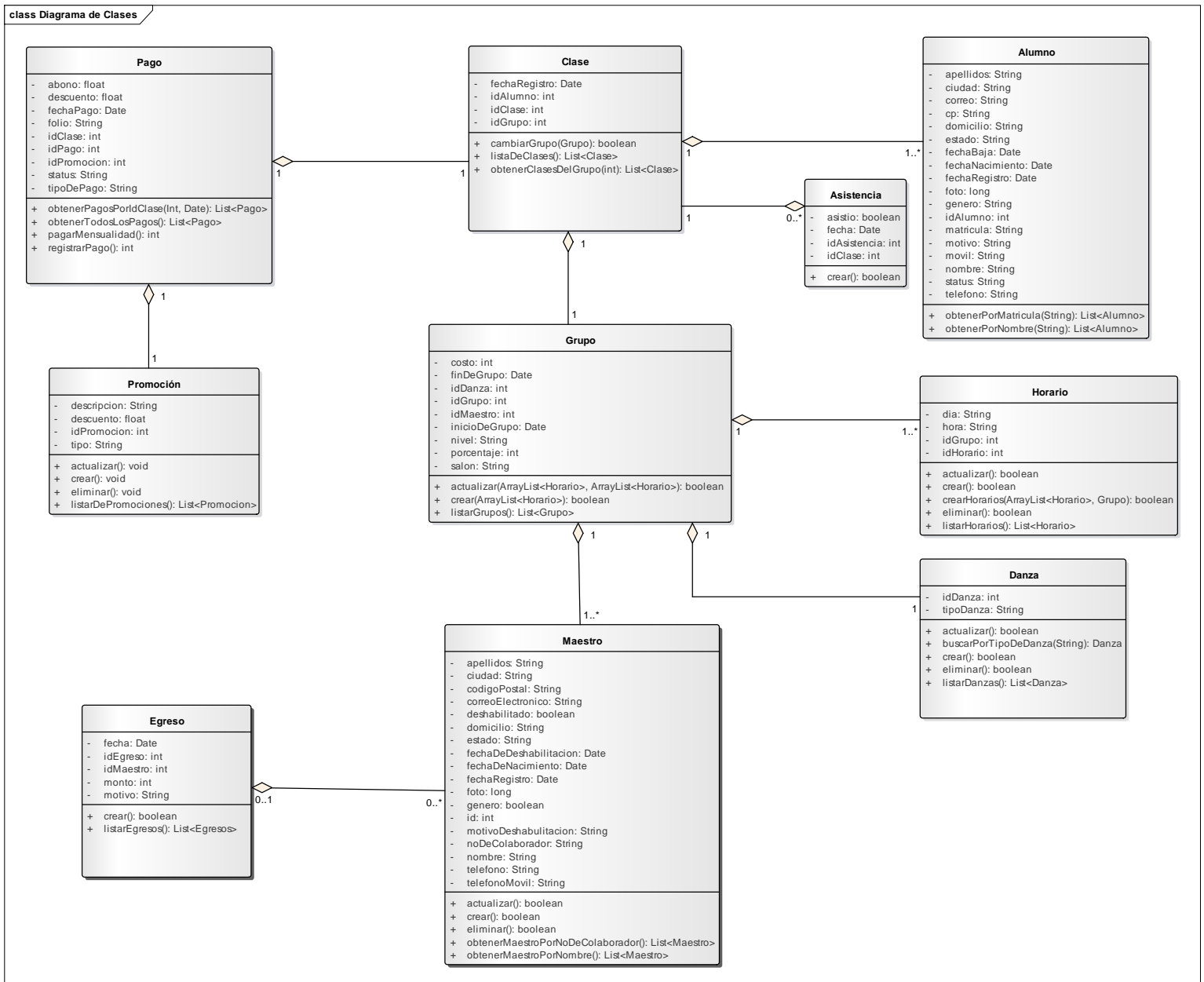
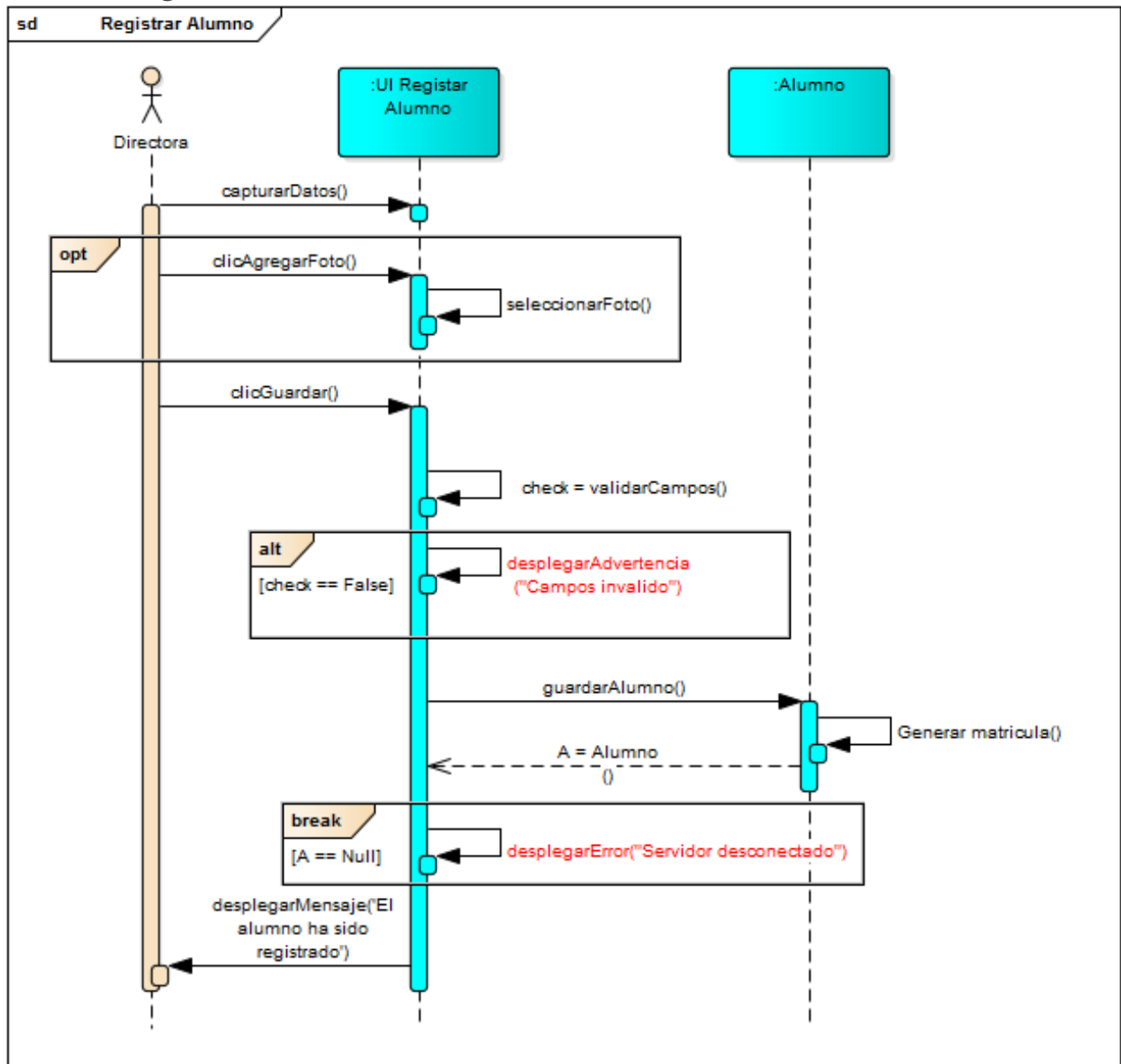


Diagrama de Clases

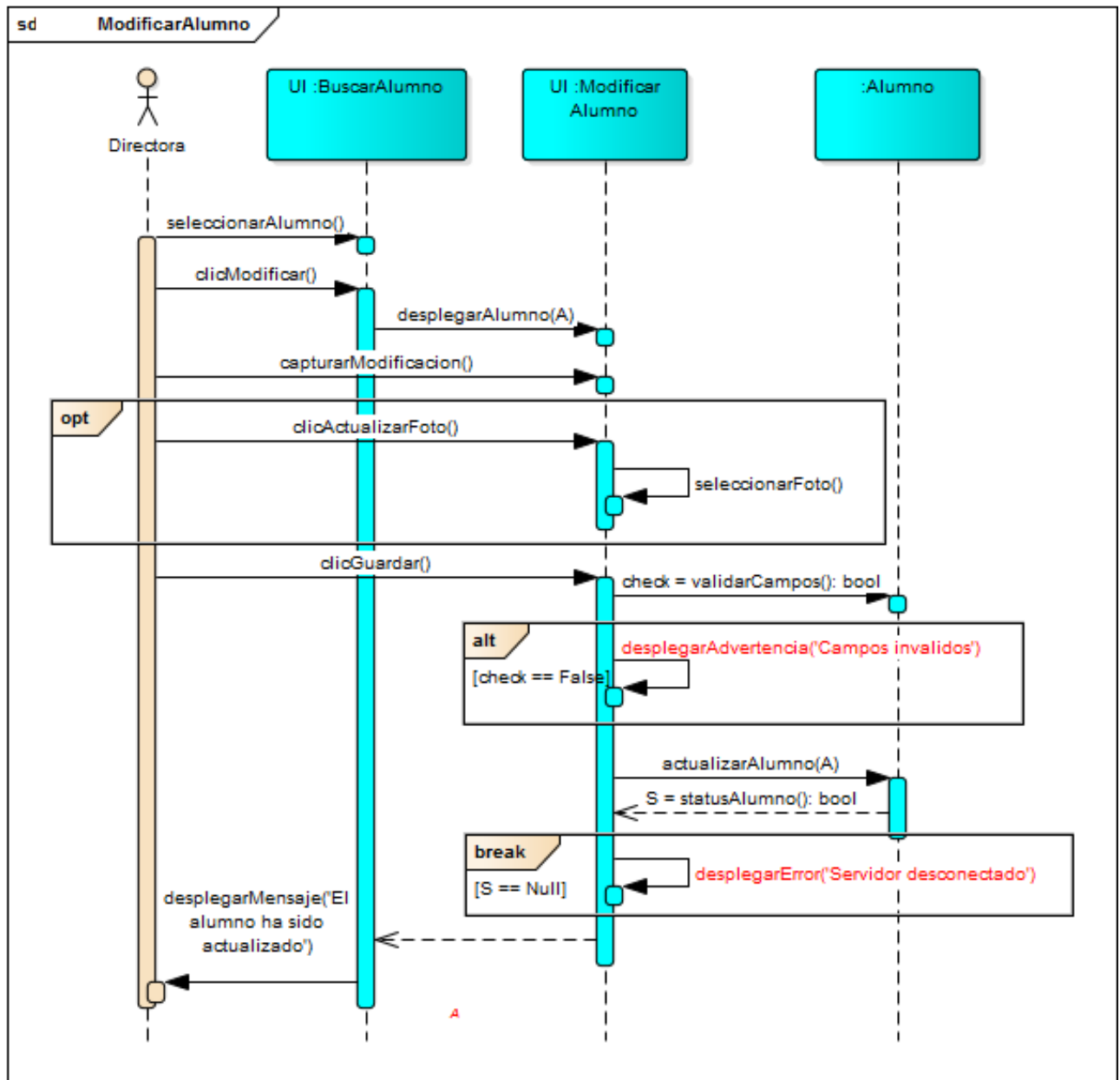


Diagramas de Secuencia

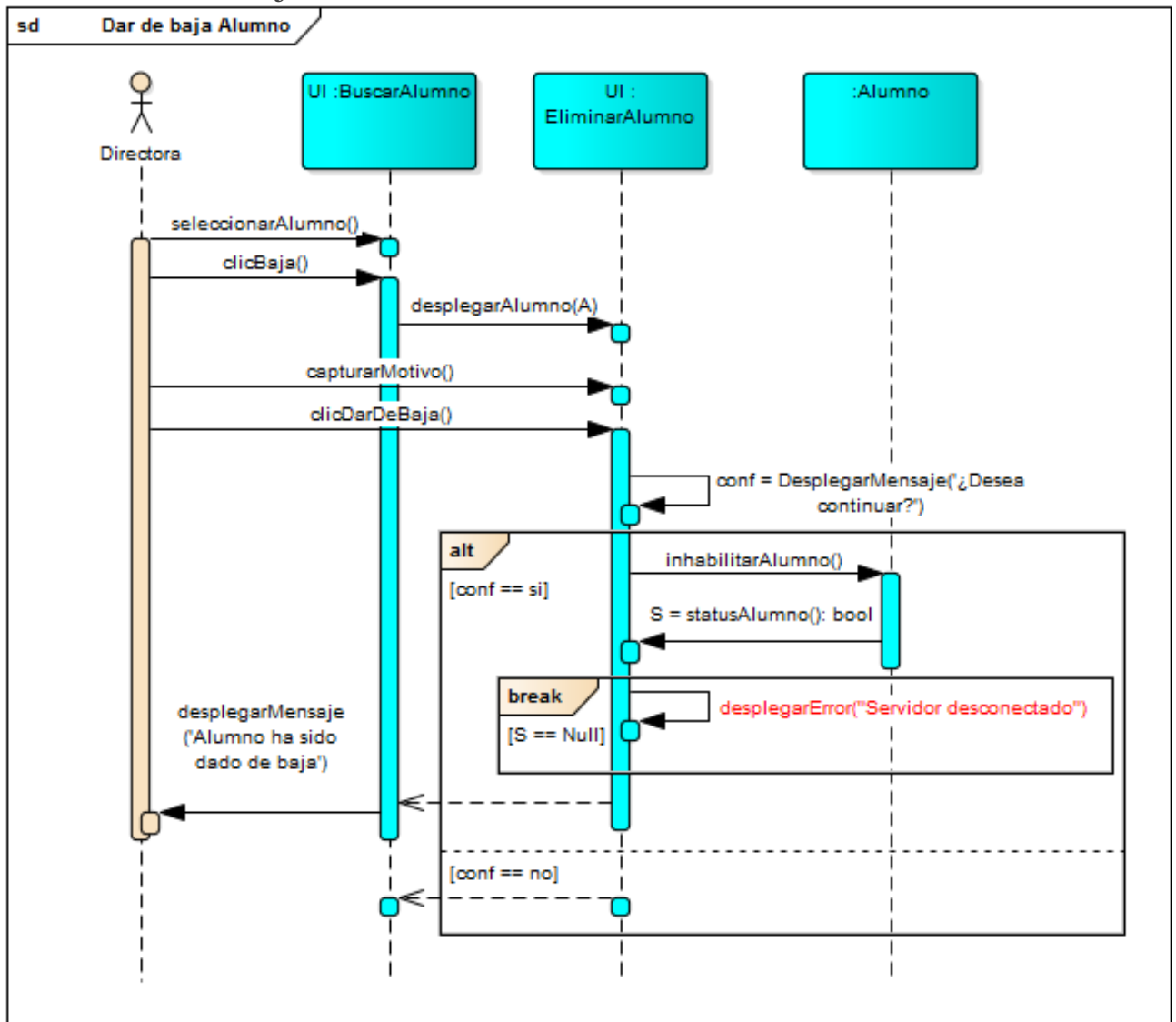
CU-01 Registrar Alumno



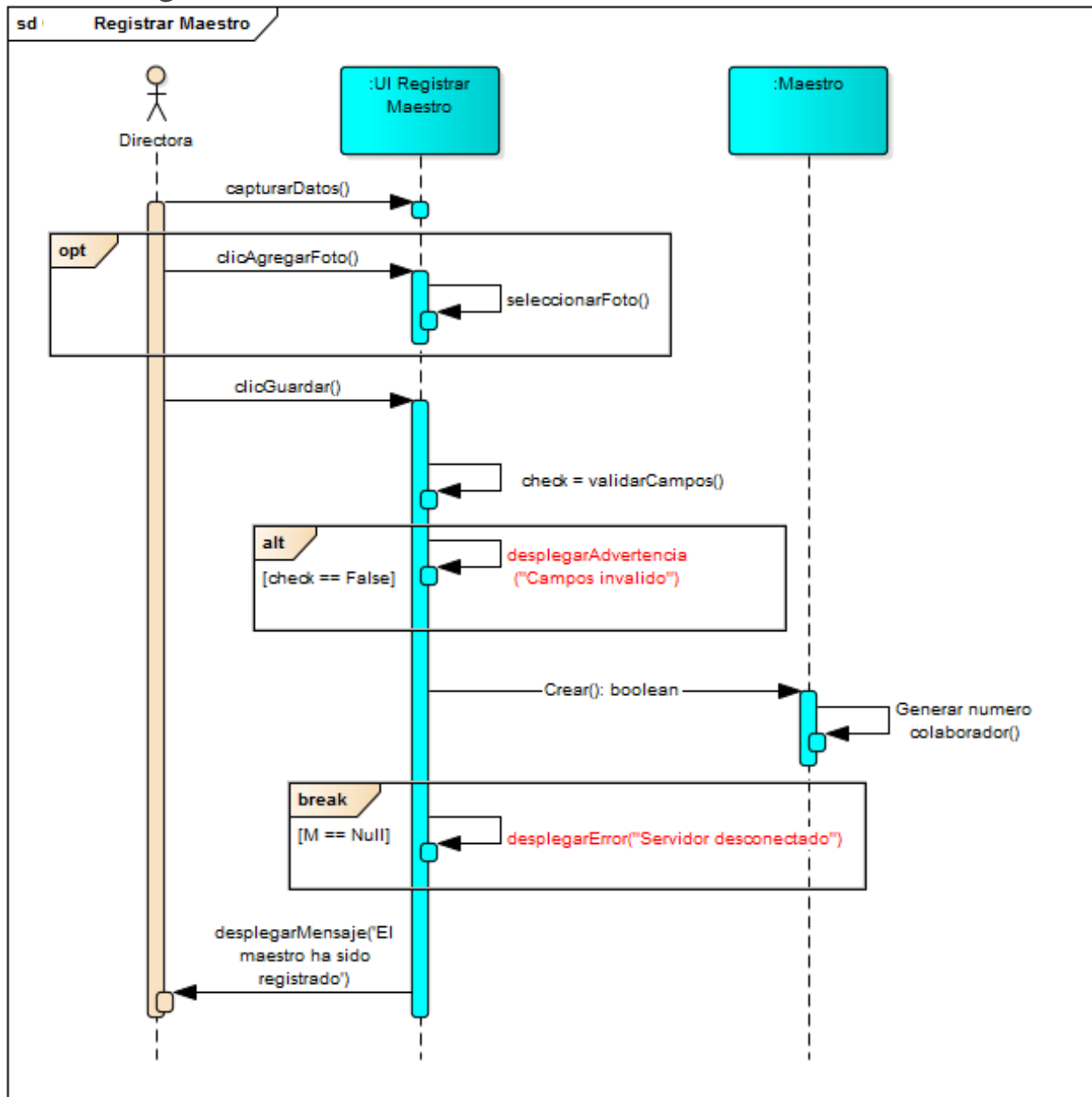
CU-02 Modificar Alumno



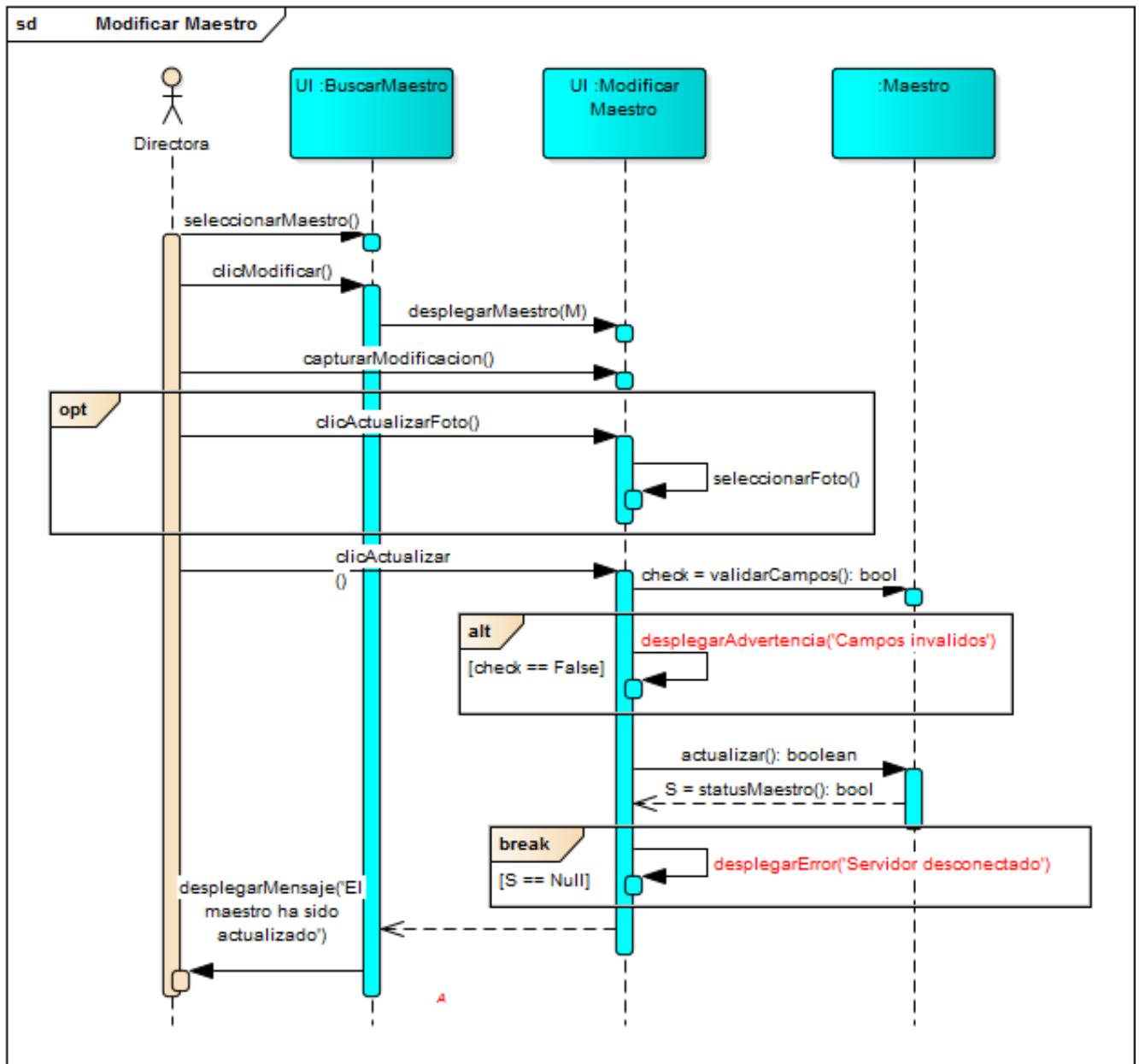
CU-03 Dar de Baja Alumno



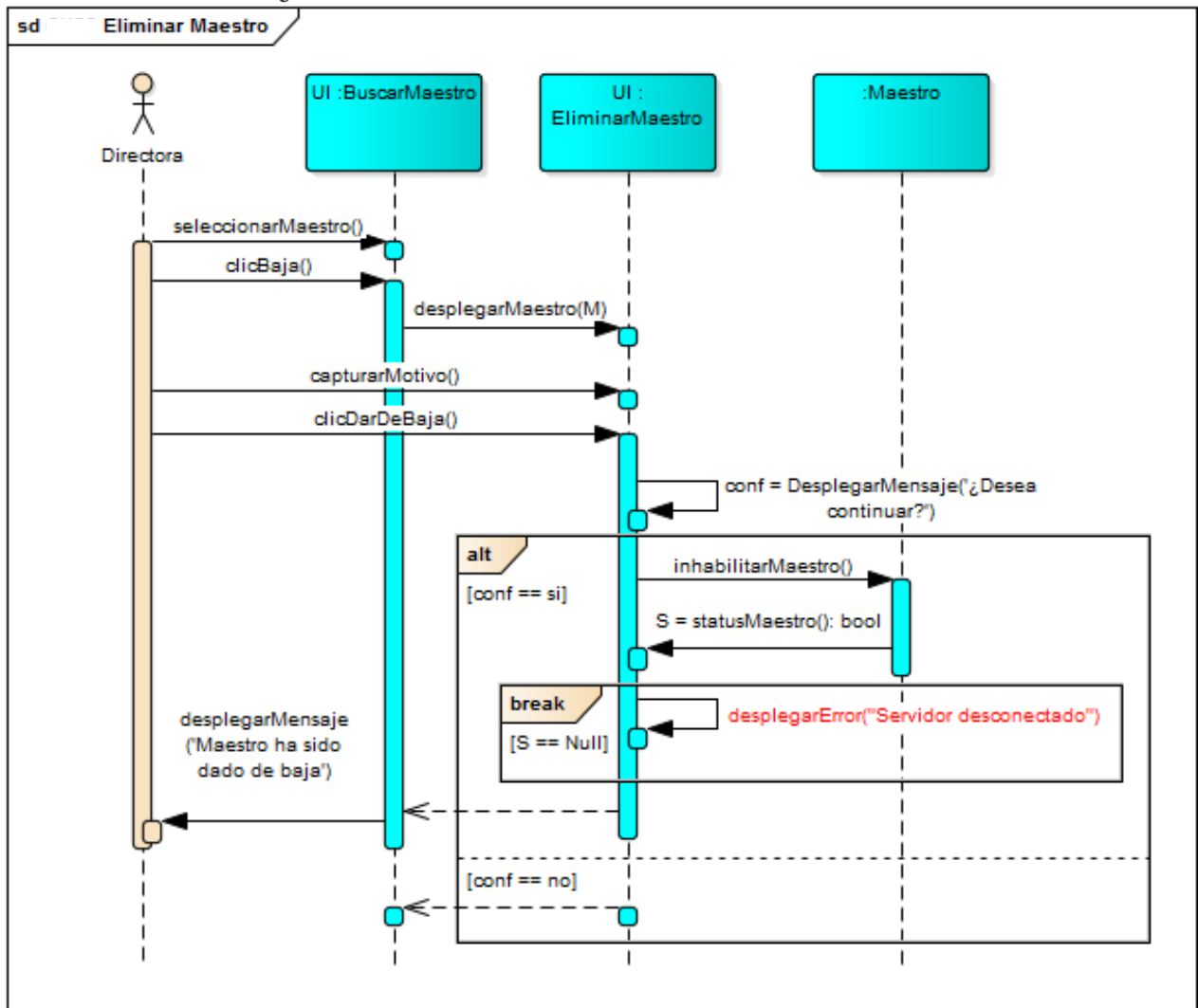
CU-04 Registrar Maestro



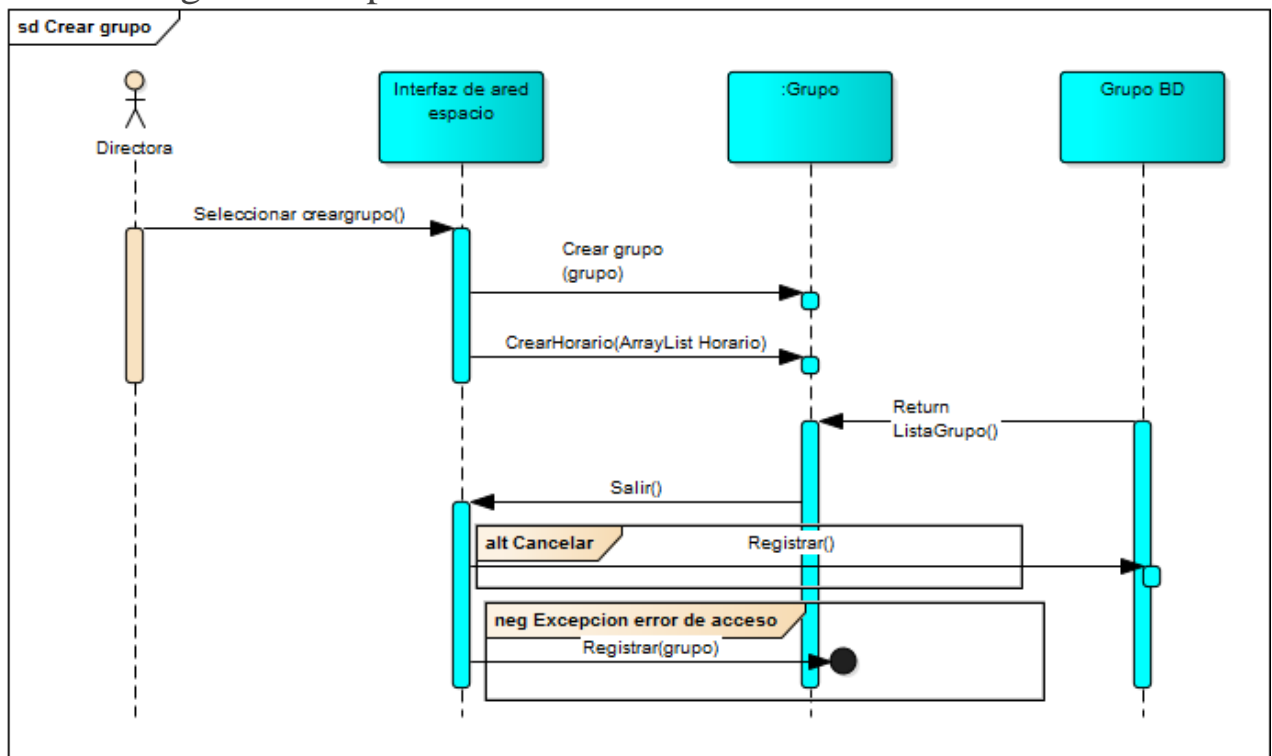
CU-05 Modificar Maestro



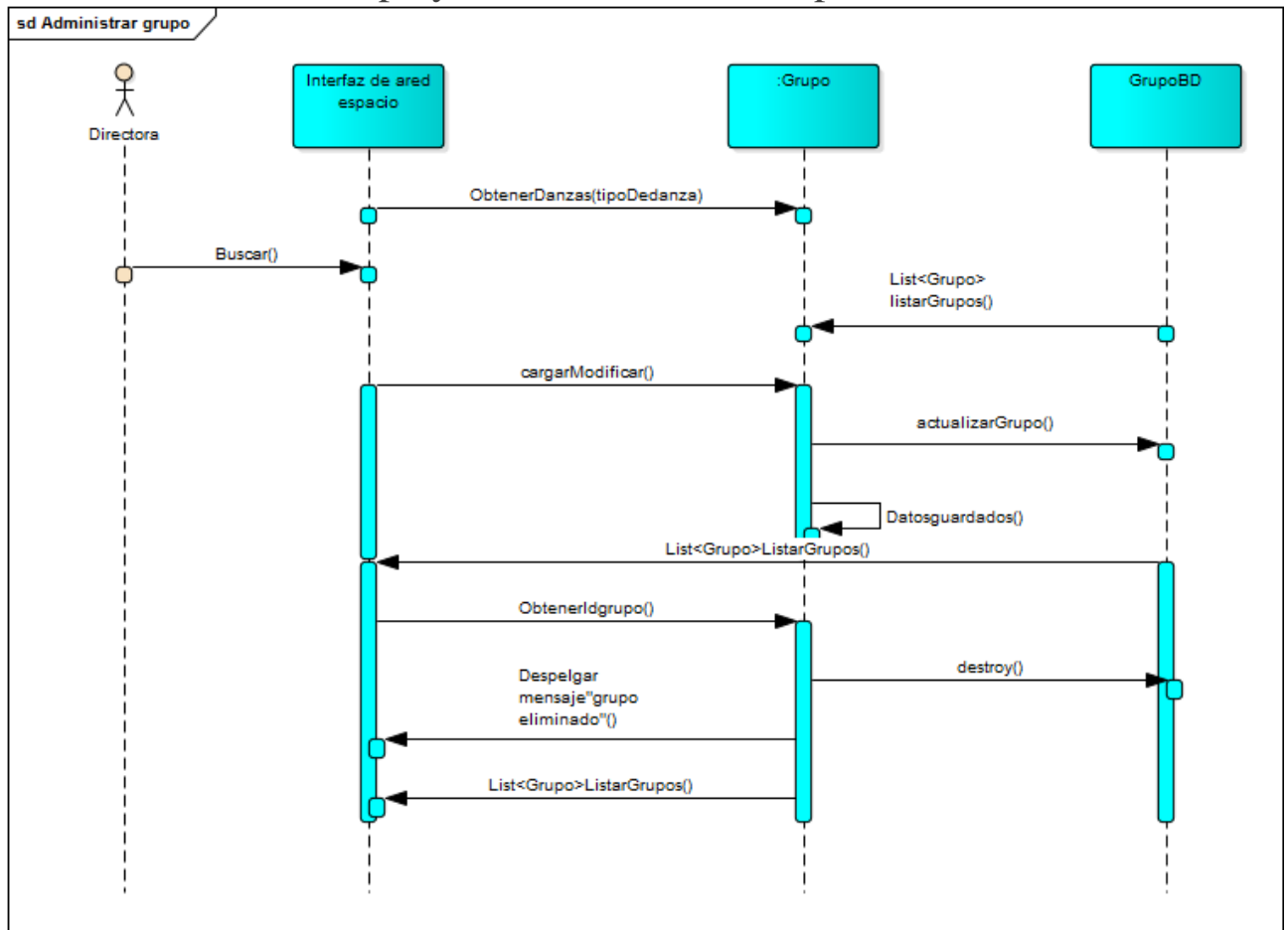
CU-06 Dar de baja Maestro



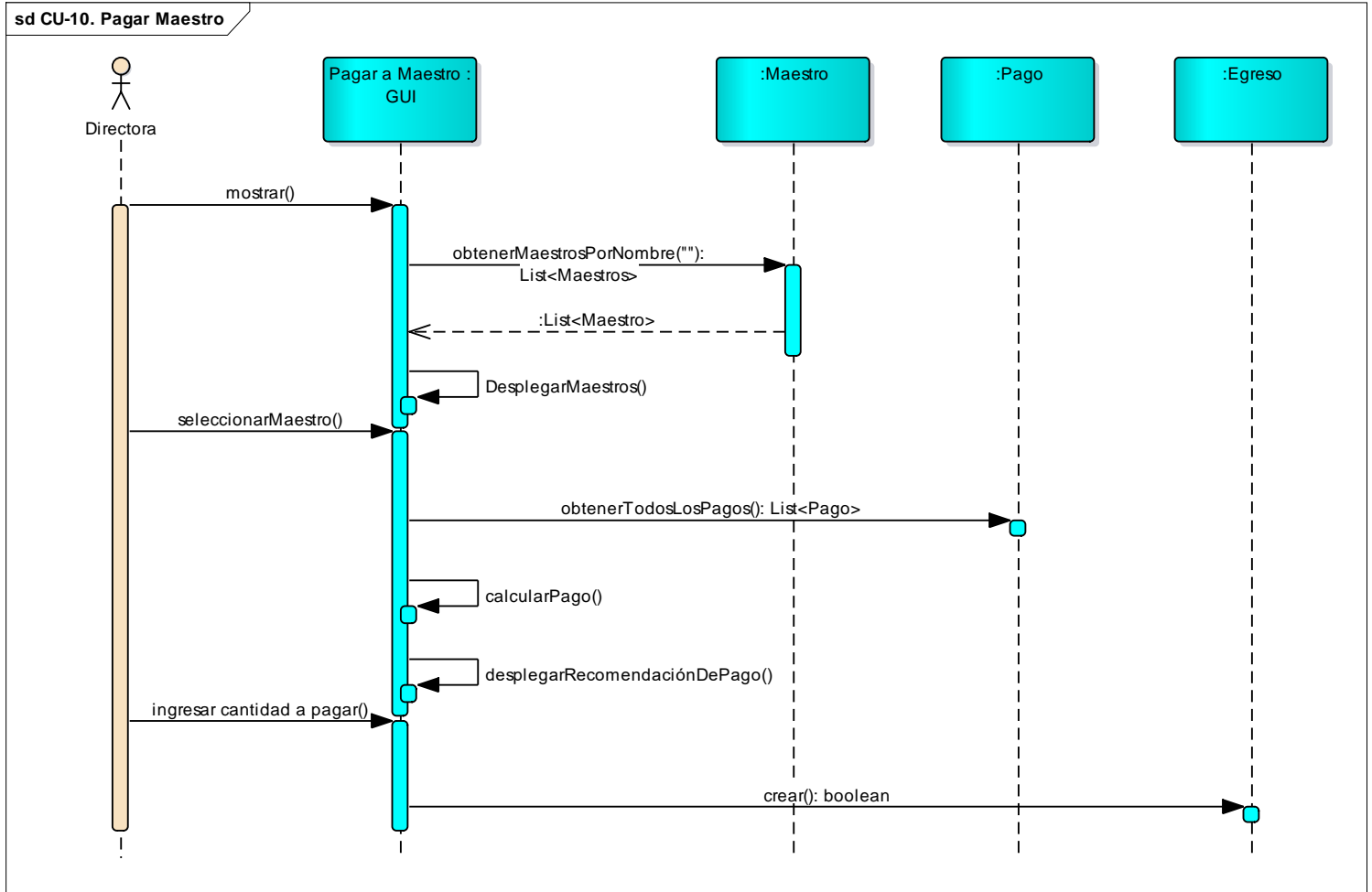
CU-07 Registrar Grupo



CU-08 Modificar Grupo y CU-09 Eliminar Grupo

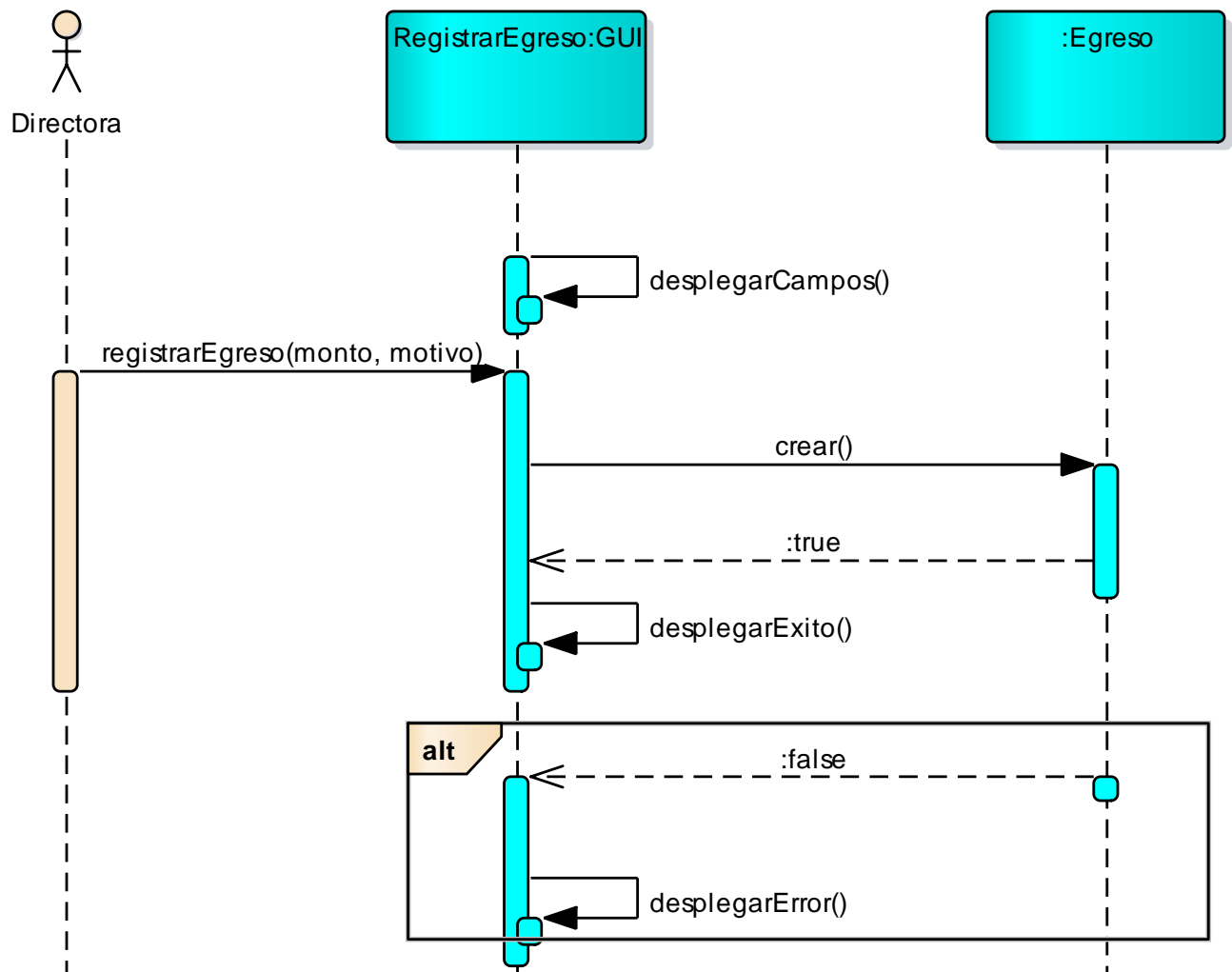


CU-10. Pagar Maestro

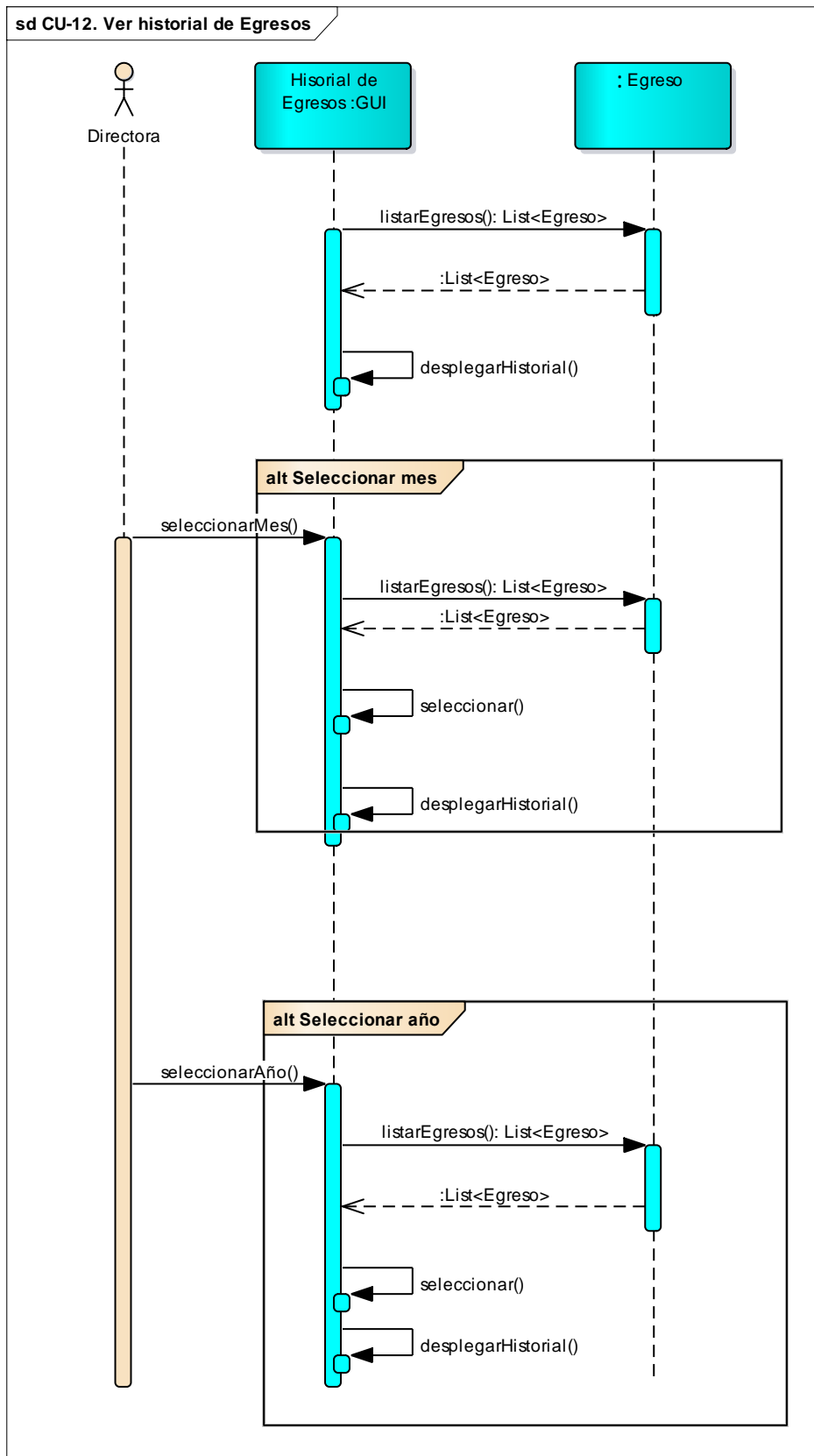


CU-11. Registrar Egreso

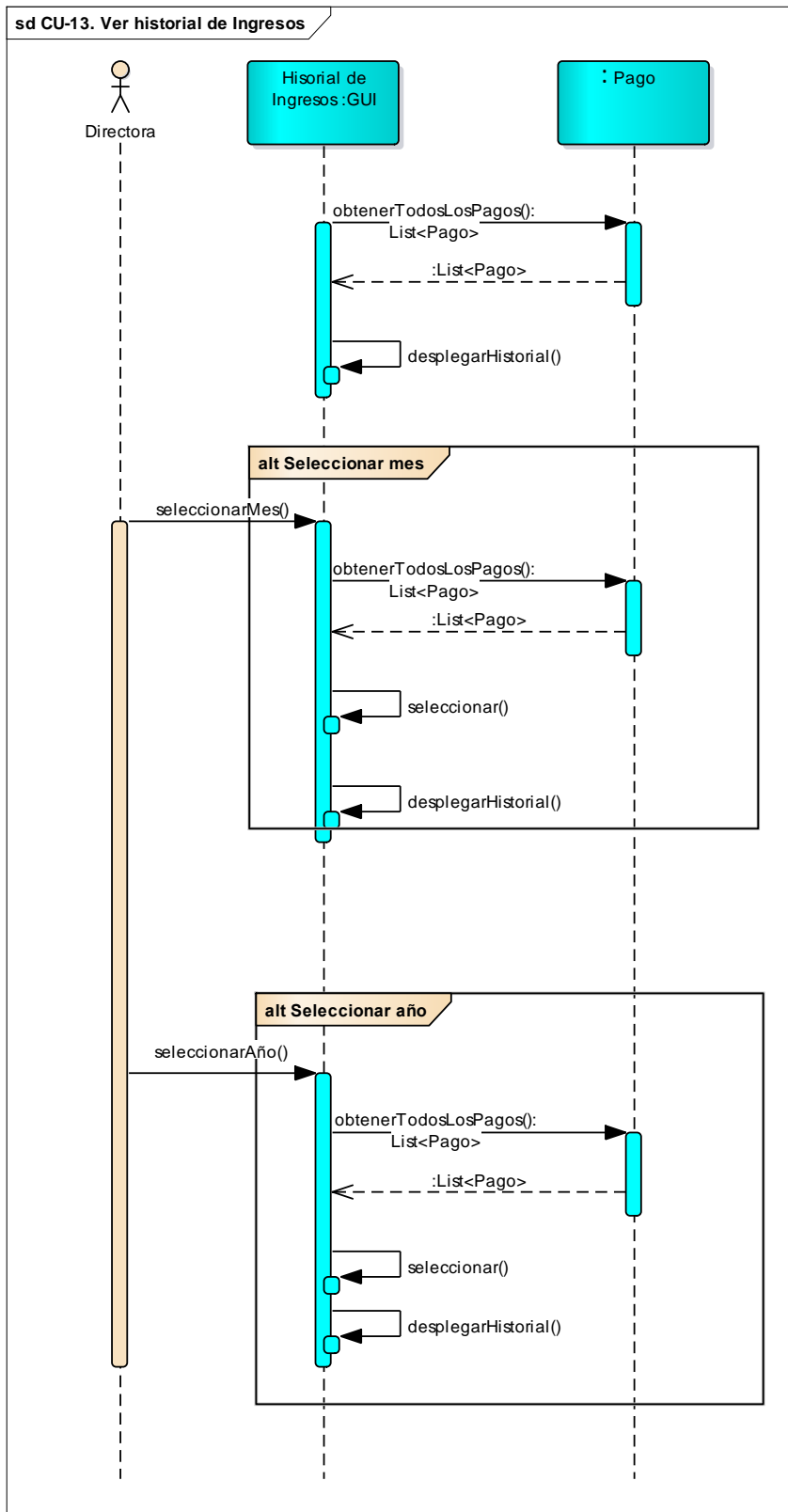
sd CU-11. Registrar Egreso



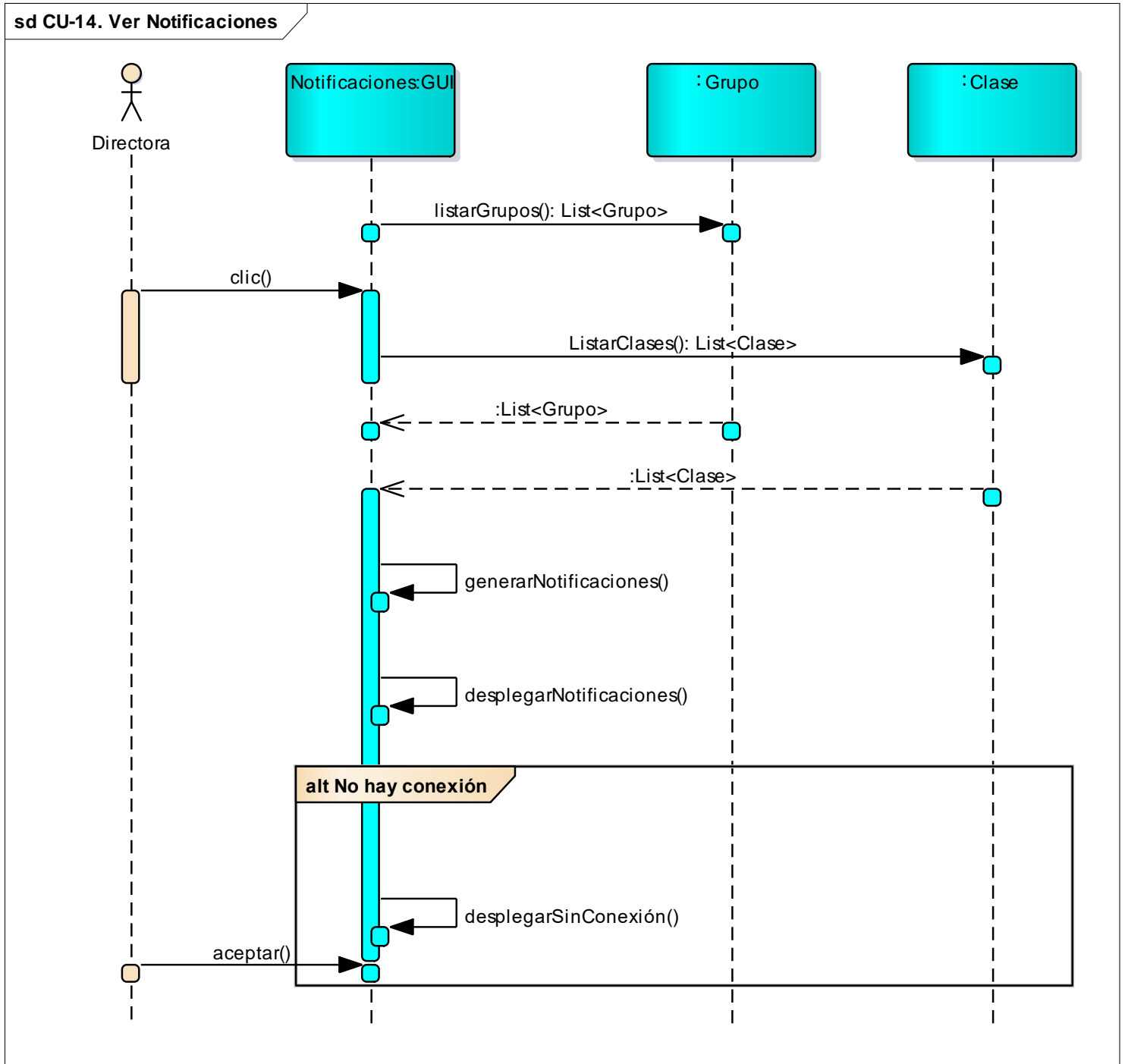
CU-12. Ver historial de Egresos



CU-13. Ver historial de Ingresos

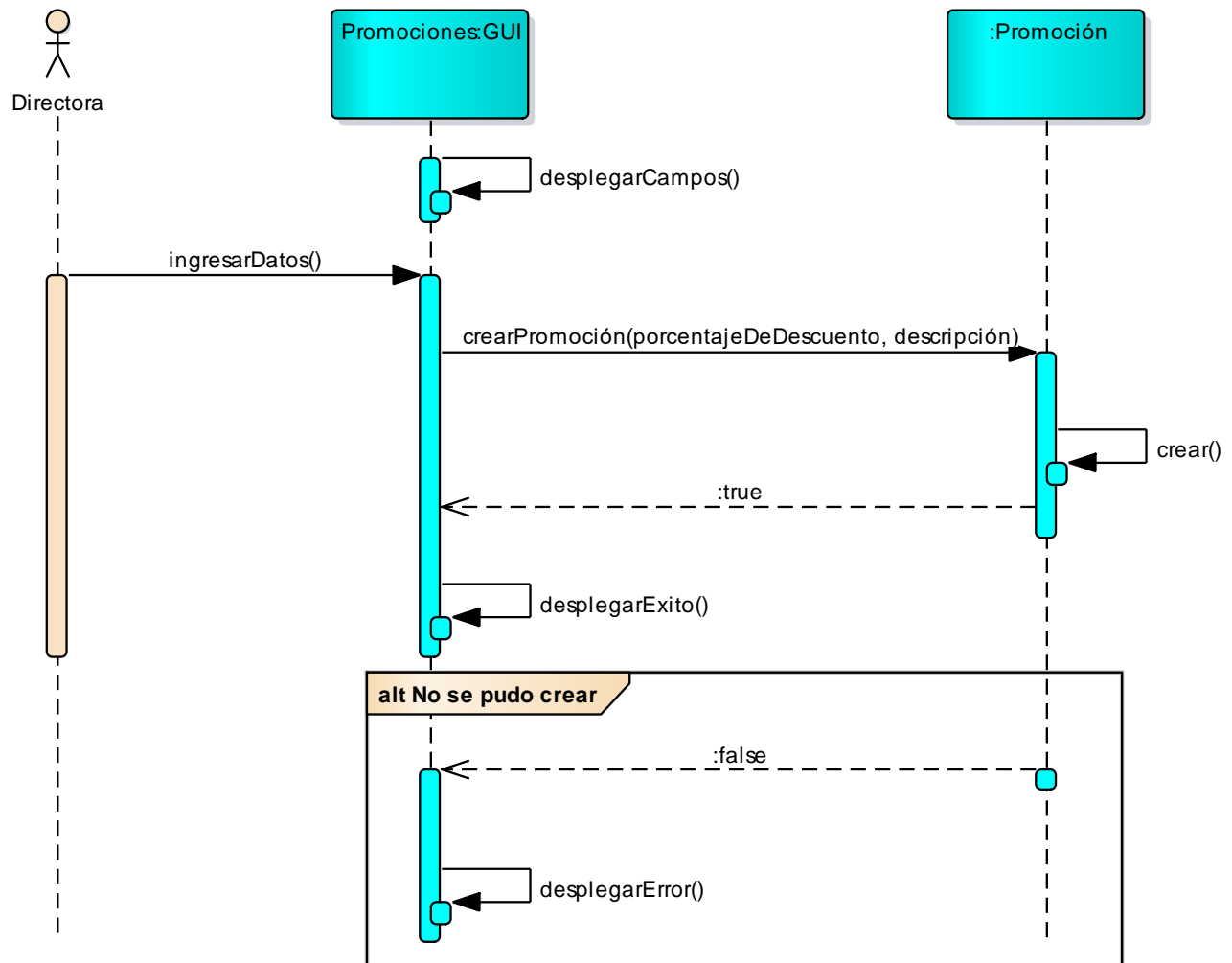


CU-14. Ver Notificaciones

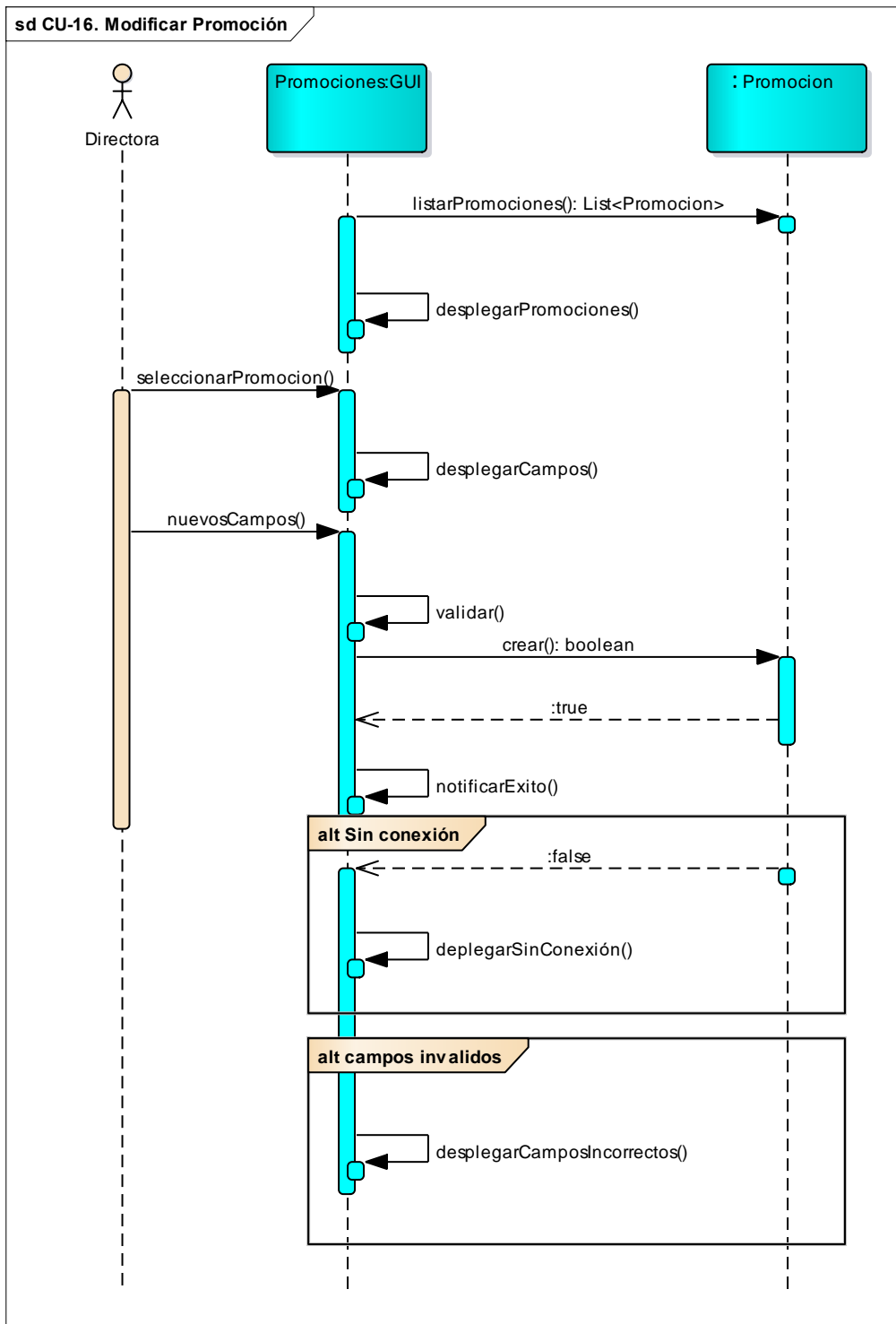


CU-15. Crear Promoción

sd CU-15. Crear Promoción

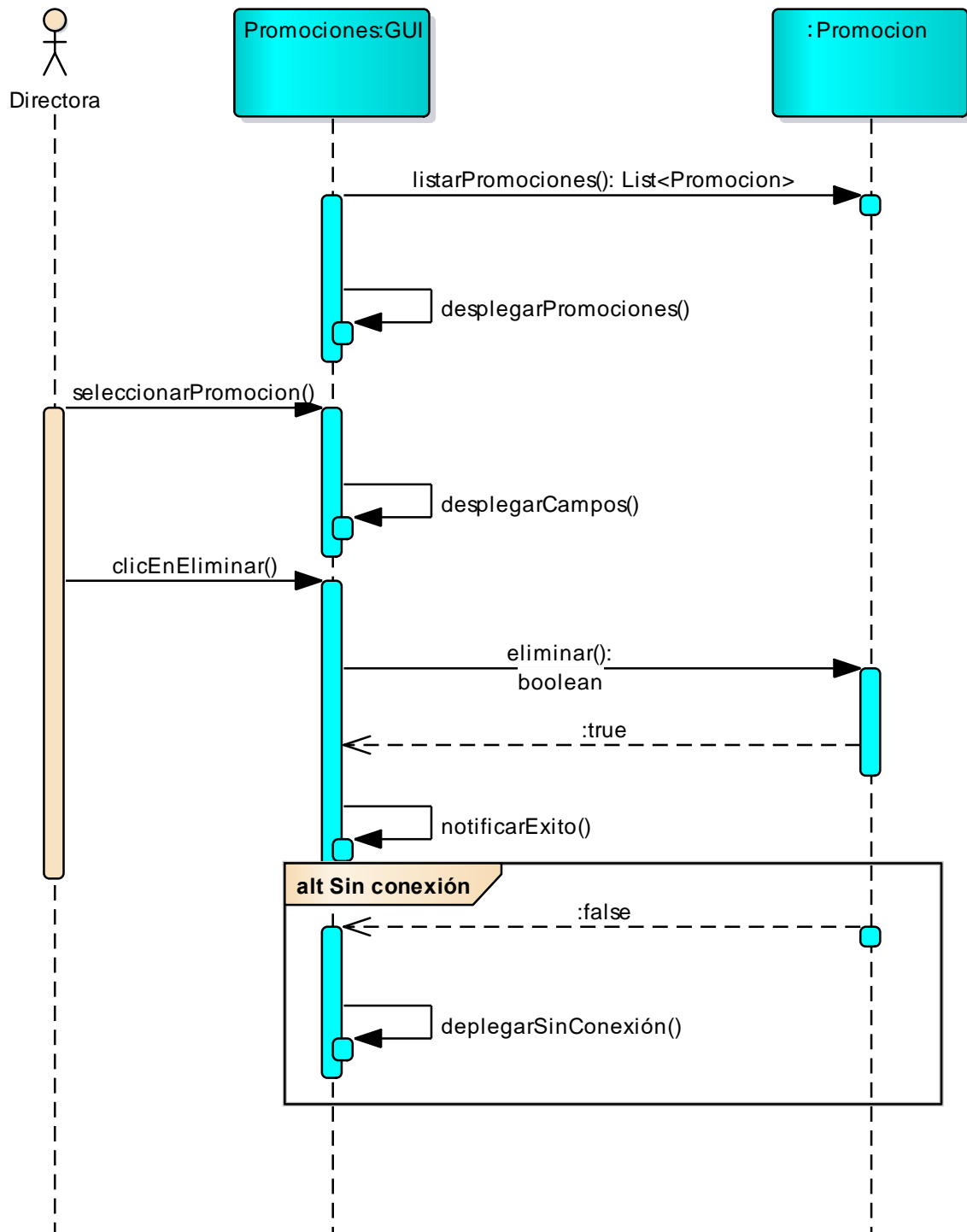


CU-16. Modificar Promoción



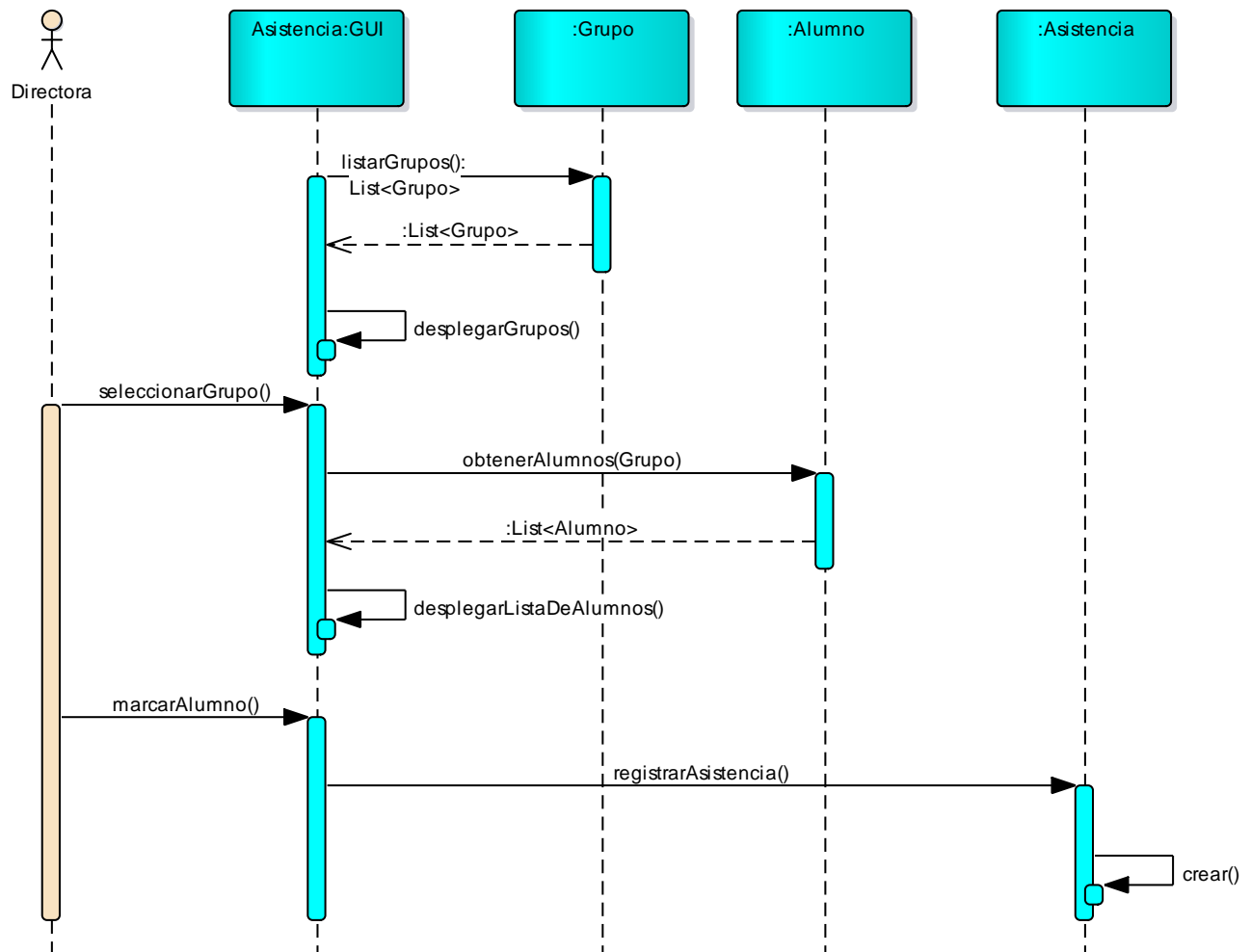
CU-17. Eliminar Promoción

sd CU-17. Eliminar Promoción

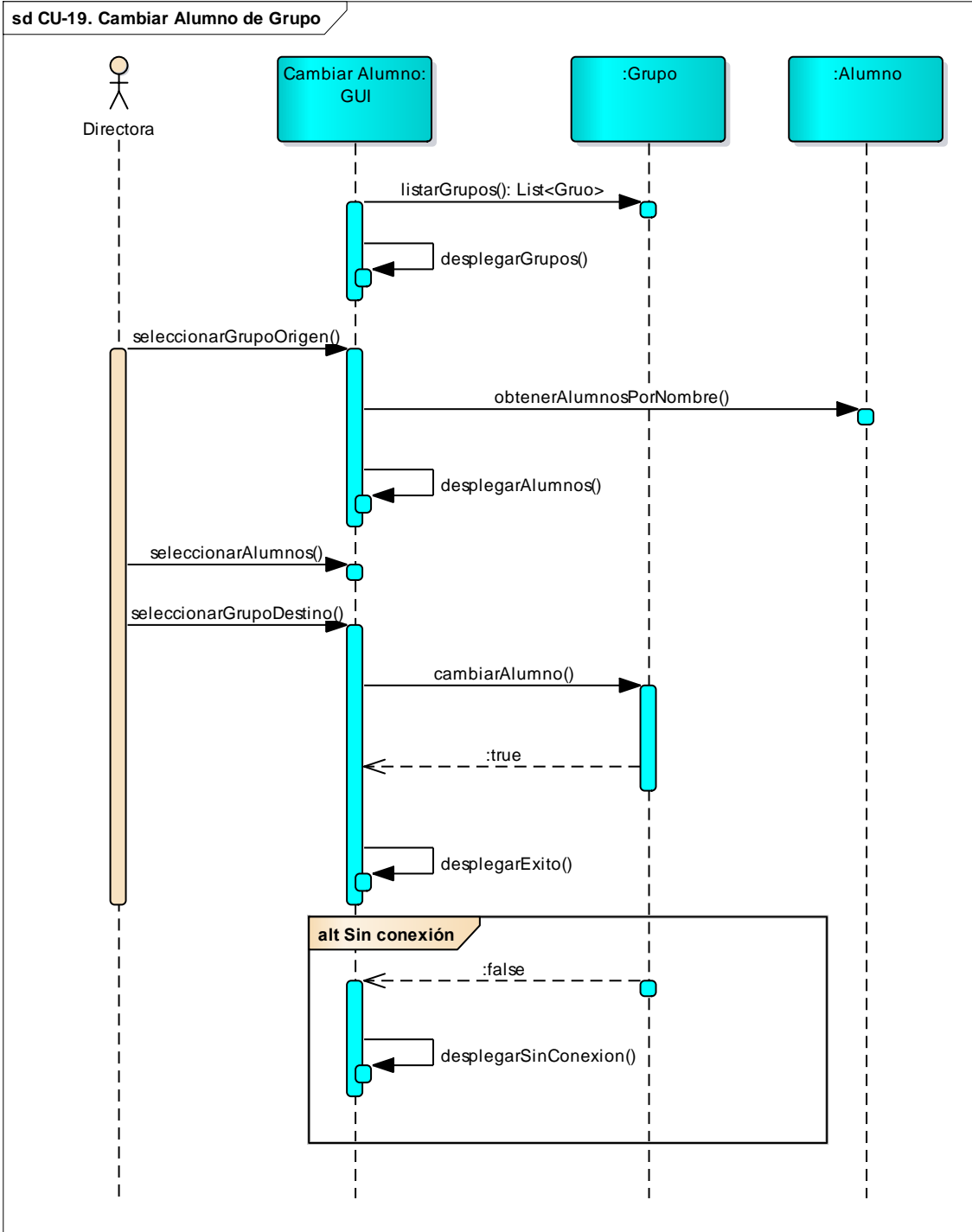


CU-18. Registrar Asistencia

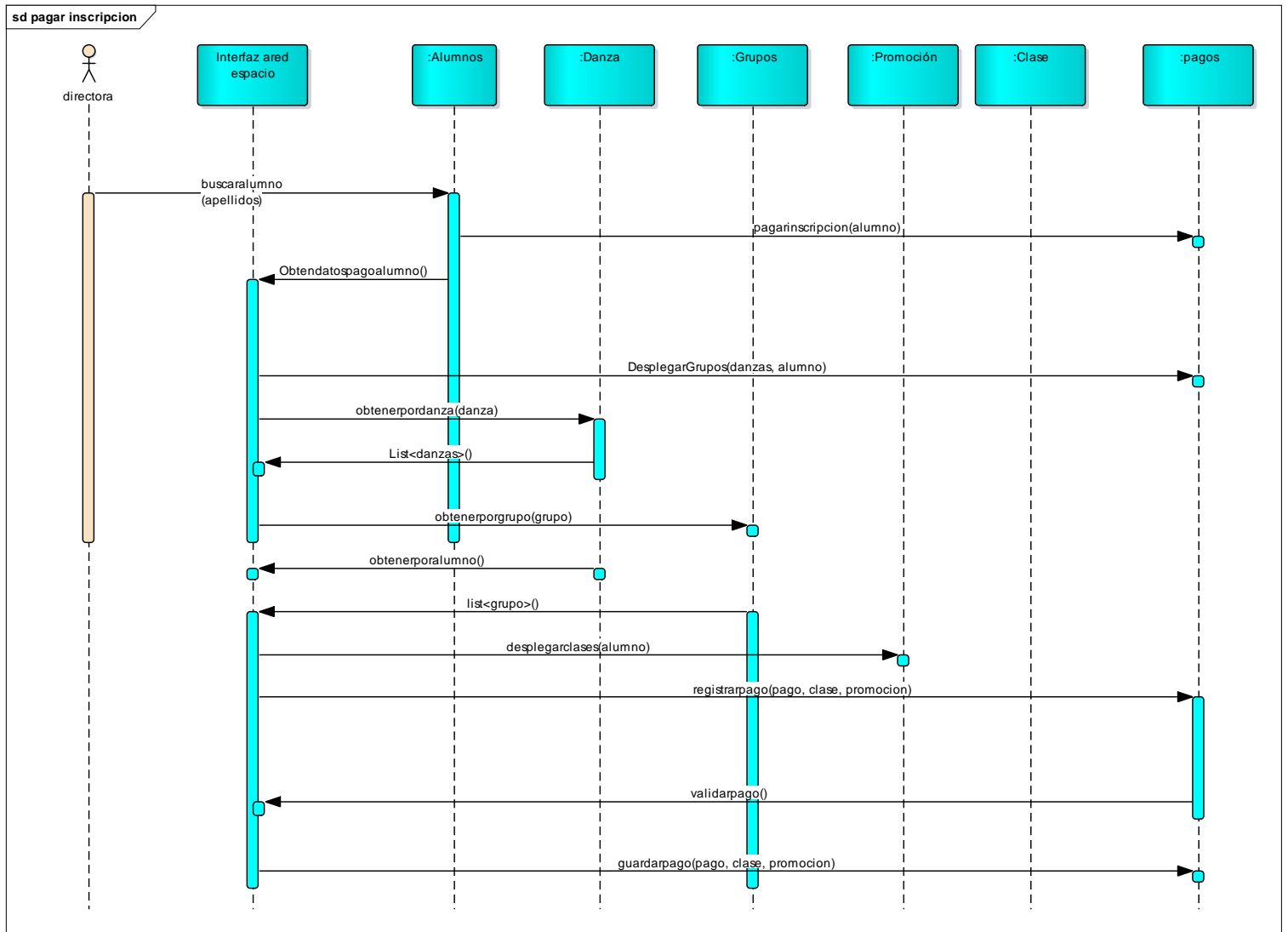
sd CU-18. Registrar Asistencia



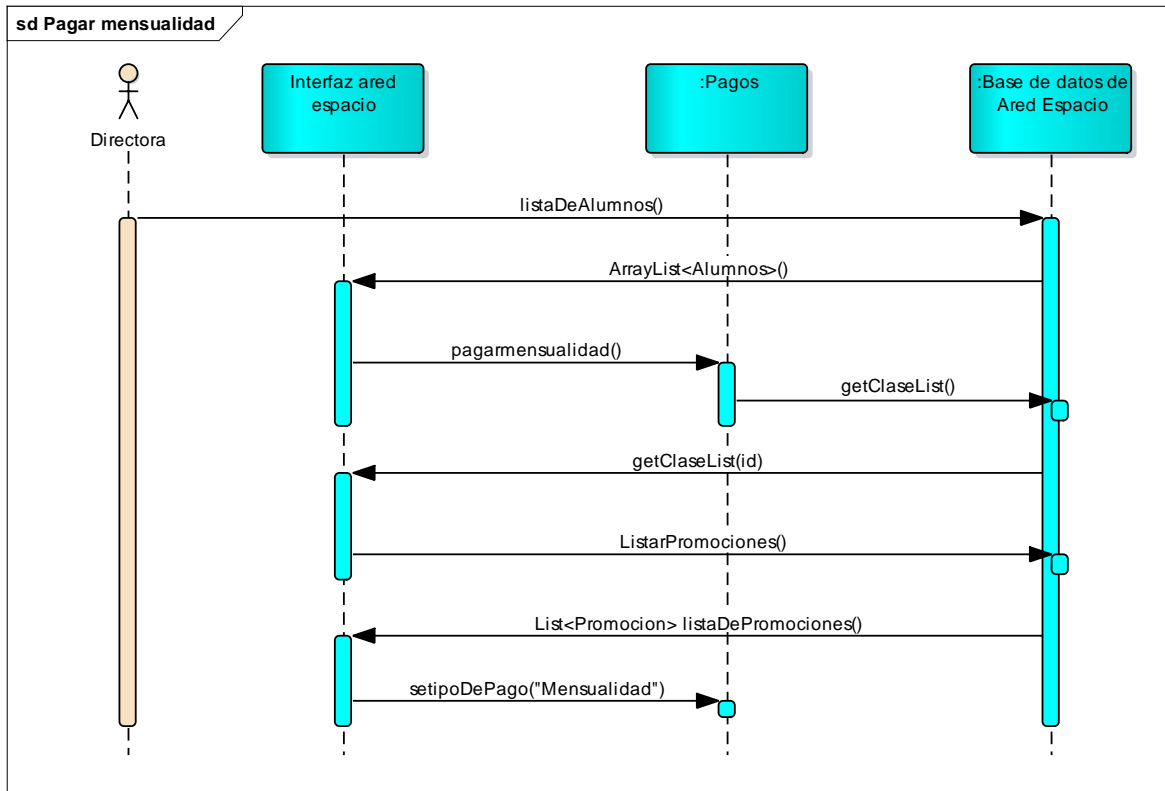
CU-19. Cambiar Alumno de Grupo



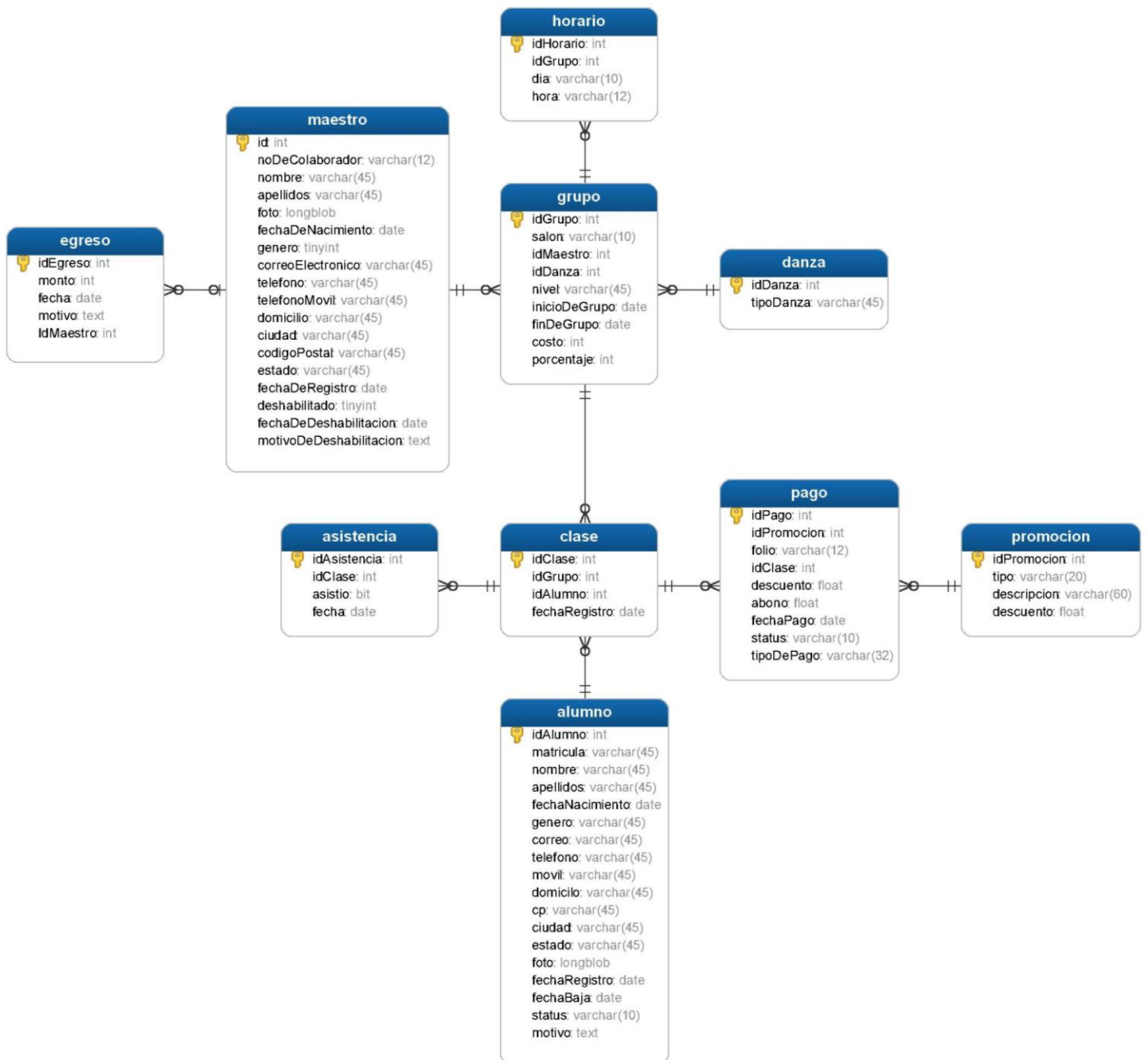
CU-20. Pagar Inscripción



CU-21. Pagar Mensualidad



Modelo Relacional



Plan de Pruebas

Clase Alumno

| Función: obteneCoincidenciasDeAlumnoPorApellido | | | | |
|-------------------------------------------------|----------------------------------------------------|--------------------------------------------|----------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Apellido="Sanchez" | El apellido existe en la base de datos de alumnos. | El sistema regresa la lista de estudiantes | El sistema regresa los datos de los estudiantes en las tablas. | V |
| Apellidos="Okumura" | El apellido no existe en la base de datos. | El sistema no regresa nada. | El sistema no regresa nada en las tablas | V |

| Función: obtenerCoincidenciasPorMatricula | | | | |
|-------------------------------------------|-----------------------------------------------------------------|--------------------------------------------|---------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| matricula="01"; | La matrícula existe en la base de datos de alumnos registrados. | El sistema regresa la lista de estudiantes | El sistema regresa los estudiantes en las tablas. | V |
| Matricula="0123456789"; | La matrícula no existe en la base de datos de alumnos. | El sistema no regresa nada. | El sistema no regresa nada en las tablas | V |

Clase Maestro

| Función: ActualizarMaestro | | | | |
|--------------------------------------|------------------------------------------|--------------------------------------------------|---------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| obtenerMaestroPorApellido("Sanchez") | El maestro existe en la base de datos | Los datos del maestro seleccionado son editables | El sistema manda el mensaje de que se actualizo correctamente | V |
| obtenerMaestroPorApellido("Hoyos") | El maestro no existe en la base de datos | La tabla de datos de | El sistema manda | V |

| | | | | |
|--|--|---------------------------|-----------------------------------------------------------|--|
| | | maestro se quedan vacios. | mensaje de que el maestro no existe o el dato es erróneo. | |
|--|--|---------------------------|-----------------------------------------------------------|--|

| Función: Registrar Maestro | | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|---------------------------------------------------|----------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Nombre("Pedro"); Apellidos("Sanchez"); Ciudad("Xalapa"); CodigoPostal("91315"); CorreoElectronico("pedro_12345@hotmail.com"); Deshabilitado(false); Domicilio("Av. Xalapa #6"); Estado("Veracruz"); Foto(null); Genero(true); Telefono("1234567890"); TelefonoMovil("1234567890"); | Los campos correctos para registrar a un maestro | El maestro aparece registrado en la base de datos | Mensaje de éxito al registrar al maestro | V |
| Nombre("213123"); Apellidos("true"); Ciudad("Xalapa"); CodigoPostal("Sanchez"); CorreoElectronico("noselaneta@hotmail.com"); Deshabilitado(true); Domicilio("Av. Xalapa #6"); Estado("Veracruz"); Foto(null); Genero(false); Telefono("abcdefgh"); TelefonoMovil("1234567890"); | Los datos de los campos a ingresar tienen valores incorrectos. | El sistema manda un mensaje de advertencia. | El sistema manda un mensaje de que se debe verificar los datos de los campos para poder Registrar. | V |
| Nombre(""); Apellidos(""); Ciudad(""); CodigoPostal(""); CorreoElectronico(""); Deshabilitado(false); Domicilio(""); Estado(""); | Los campos vacíos para registrar un maestro | Mensaje de advertencia | El sistema despliega un mensaje de advertencia, de que los | V |

| | | | | |
|---------------------------------------------------------------------|--|--|---------------------------|--|
| Foto(null); Genero(true); Telefono(""); TelefonoMovil(""); | | | campos están vacíos | |
|---------------------------------------------------------------------|--|--|---------------------------|--|

| Función: obtenerMaestroPorNoDeColaborador | | | | |
|-------------------------------------------|----------------------------------------------------|--------------------------------------------|---------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| noDeColaborador=2213 | El apellido existe en la base de datos de alumnos. | El sistema regresa la lista de estudiantes | El sistema regresa los estudiantes en las tablas. | V |
| noDeColaborador=000 | El apellido existe en la base de datos de alumnos. | El sistema no regresa nada. | El sistema no regresa nada en las tablas | V |

| Función: obtenerMaestroPorNombre | | | | |
|----------------------------------|--------------------------------------------------|--------------------------------------------|---------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| nombre=Pedro | El nombre existe en la base de datos de alumnos. | El sistema regresa la lista de estudiantes | El sistema regresa los estudiantes en las tablas. | V |
| nombre=pánfilo | El nombre existe en la base de datos de alumnos. | El sistema no regresa nada. | El sistema no regresa nada en las tablas | V |

| Función: obtenerMaestroPorApellidos | | | | |
|-------------------------------------|----------------------------------------------------|--------------------------------------------|---------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Apellidos=Sanchez | El apellido existe en la base de datos de alumnos. | El sistema regresa la lista de estudiantes | El sistema regresa los estudiantes en las tablas. | V |
| Apellidos=Hoyos | El apellido existe en la base de datos de alumnos. | El sistema no regresa nada. | El sistema no regresa nada en las tablas | V |

Clase Grupo

| Función: CrearGrupo | | | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>Nivel("Basico"); Salon("CC3"); TipoDeDanza("Ballet"); Maestro maestro = Maestro.obtenerMaestroPorApellido("").get(0); grupo.setIdMaestro(maestro); ArrayList<Horario> horarios = new ArrayList<>(); boolean resultado = grupo.crear(horarios);</pre> | El grupo no existe en la base de datos. | El grupo se verá reflejado en la tabla. | El sistema manda un mensaje de que el grupo fue creado exitosamente. | V |
| <pre>setNivel(""); setSalon(""); grupo.setTipoDeDanza(""); Maestro maestro = Maestro.obtenerMaestroPorApellido("").get(0); grupo.setIdMaestro(maestro); ArrayList<Horario> horarios = new ArrayList<>(); boolean resultado = grupo.crear(horarios);</pre> | El grupo no existe en la base de datos, pero los campos están vacíos. | El sistema manda una advertencia. | El sistema manda una advertencia de que los campos están vacíos y deben verificarse. | V |

| Función: ActualizarGrupo | | | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|-----------------------------------------|----------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>Nivel("Basico"); Salon("CC3"); TipoDeDanza("Ballet"); Maestro maestro = Maestro.obtenerMaestroPorApellido("").get(0); grupo.setIdMaestro(maestro); ArrayList<Horario> horarios = new ArrayList<>(); boolean resultado = grupo.crear(horarios);</pre> | El grupo no existe en la base de datos. | El grupo se verá reflejado en la tabla. | El sistema manda un mensaje de que el grupo fue creado exitosamente. | V |
| <pre>setNivel(""); setSalon(""); grupo.setTipoDeDanza("");</pre> | El grupo no existe en la base | El sistema manda una | El sistema manda una advertencia | V |

| | | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|--------------|-----------------------------------------------------|--|
| <pre> Maestro maestro = Maestro.obtenerMaestroPorApellido("").get(0); grupo.setIdMaestro(maestro); ArrayList<Horario> horarios = new ArrayList<>(); boolean resultado = grupo.crear(horarios); </pre> | de datos, pero los campos están vacíos. | advertencia. | de que los campos están vacíos y deben verificarse. | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|--------------|-----------------------------------------------------|--|

| Función: obtenerTodosLosGrupos | | | | |
|------------------------------------------------------------------|--------------------------------------------|---------------------------------------------|-------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| ArrayList<Grupo> grupos = new ArrayList<>(Grupo.listarGrupos()); | Hay grupos registrados en la base de datos | El sistema regresa la lista de grupos | EL sistema muestra los grupos registrados | V |
| ArrayList<Grupo> grupos = new ArrayList<>(Grupo.listarGrupos()); | Aun no hay grupos registrados. | El sistema manda un mensaje de advertencia. | El sistema manda un mensaje de que no hay algún grupo registrado. | V |

Clase Foto

| Función: obtenerImagenCorrecta | | | | |
|-----------------------------------------------------------------------|------------------------------------------------------|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| ruta = "C:\\Users\\Mauricio Juárez\\Desktop\\Imagen_TEST.jpg"; (); | La ruta de la imagen a subir está bien escrita. | El sistema obtiene la imagen correctamente de la ruta. | Se muestra la imagen elegida en la ventana. | V |
| ruta = ""; | El campo de la ruta esta vacío o no se eligió alguna | El sistema manda una advertencia. | El sistema despliega el mensaje de advertencia de que no hay alguna coincidencia con el archivo que se busco. | V |

| Función: AgregarImagenCorrecta | | | | |
|----------------------------------------------------------------|-----------------------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| ruta = "C:\\Users\\Mauricio Juárez\\Desktop\\Imagen_TEST.jpg"; | La ruta de la imagen está en el formato correcto. | El sistema obtiene la ruta correcta | Se muestra la imagen elegida en la ventana. | V |
| ruta = "file:C:Imagen_TEST.jpg"; | La ruta de la imagen está en un formato incorrecto. | El sistema manda una advertencia. | El sistema despliega el mensaje de advertencia y la venta se queda vacía, donde se mostraría la foto elegida. | V |

Clase Horario

| Función: CrearHorario | | | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre> horario.setDia("Lunes"); horario.setHora("12:00-14:00"); Grupo grupo = Grupo.listarGrupos().get(0); horario.setIdGrupo(grupo); boolean resultado = horario.crear(); </pre> | Los datos de horario de un grupo no han sido registrados en la base de datos. | El sistema muestra el horario agregado al grupo. | El sistema manda un mensaje de que el registro del grupo fue exitoso. | V |
| <pre> Horario horario = new Horario(); horario.crear(); </pre> | Los datos de horario están en blanco. | El sistema manda una advertencia. | El sistema despliega el mensaje de advertencia de que se debe ingresar datos correctos en los campos solicitados. | V |

| Función: ActualizarHorario | | | | |
|--------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Horario horario = grupo.getHorarioList().get(0); horario.setDia("Martes"); boolean resultado = horario.actualizar(); | El horario del grupo solicitado está registrado en la base de datos. | El sistema muestra los datos del horario y los vuelve editables. | El sistema manda un mensaje de que el horario se actualizo exitosamente. | V |
| Horario horario = new Horario(); horario.actualizar(); | El horario del grupo solicitado es incorrecto o no existe. | Las tablas de horarios se quedan vacias. | El sistema despliega el mensaje de advertencia de que el horario del grupo solicitado, es incorrecto o no existe. | V |

| Función: EliminarHorario | | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------|--------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Grupo grupo = Grupo.listarGrupos().get(0); Horario horario = grupo.getHorarioList().get(0); boolean resultado = horario.eliminar(); | El horario solicitado existe en la base de datos. | El sistema elimina el horario seleccionado de la base de dstos. | El sistema muestra un mensaje de horario eliminado satisfactoriamente. | V |
| Horario horario = new Horario(); horario.eliminar(); | El horario solicitado no existe en la base de datos | Las tablas se quedan vacías. | El sistema despliega el mensaje de advertencia donde indica que el horario solicitado no existe. | V |

Clase Asistencia

| Función: Crear Asistencia | | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>Asistencia asistencia = new Asistencia(); asistencia.setAsistio(true); asistencia.setFecha(new Date(new GregorianCalendar().getTimeInMillis())); asistencia.setIdClase(Clase.listaDeClases().get(0)); boolean resultado = asistencia.crear(); Assert.assertTrue("Prueba registrar asistencia correctamente", resultado); }</pre> | Se selecciona un grupo y el pase de asistencia esta limpio o libre. | El sistema registra la fecha y marca que un alumno asistió ese día, | El sistema marca la casilla del nombre seleccionado con un circulo negro y manda un mensaje de registro de asistencia exitoso. | V |
| <pre>Asistencia asistencia = new Asistencia(); boolean resultado = asistencia.crear(); Assert.assertTrue("Prueba registrar asistencia incorrecta", resultado);</pre> | El pase de asistencia esta libre y no se toma la lista de alumnos de una clase. | No se toma la fecha, ni la lista de alumnos de una clase. | El sistema manda un mensaje de que no hay conexión a la base de datos y no muestra nada. | V |

Clase Danza

| Función: ObtenerTodasLasDanzas | | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|--------------------------------------------------------------------|---------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>List<Danza> lista = Danza.obtenerTodas(); boolean resultado = !lista.isEmpty(); assertTrue("Prueba obtener todas las danzas", resultado);</pre> | Se manda la cadena vacia para buscar todas las danzas. | El sistema regresa todas los tipos de danzas registradas en la bd. | El sistema muestra en la tabla de danzas, todos los tipos de danza registrados. | V |

| Función: CrearDanza | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>Danza danza = new Danza(); danza.setTipoDanza("Prueba Crear danza"); boolean resultado = danza.crear(); assertTrue("Prueba crear danza correctamente", resultado); }</pre> | Se mandan los datos para crear la danza correctamente | El sistema registra la danza en la base de datos | El sistema muestra la danza registrada en la bd y manda mensaje de que fue registrada correctamente | V |
| <pre>Danza danza = new Danza(); boolean resultado = danza.crear(); assertTrue("Prueba crear danza incorrectamente", resultado);</pre> | Se manda los datos cuando no hay conexión o se manda incorrectamente. | El sistema no registra nada. | El sistema manda un mensaje de que hubo un error de conexión. | V |

| Función: ActualizarDanza | | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>Danza danza = Danza.obtenerTodas().get(0); danza.setTipoDanza("Prueba actualizar danza"); boolean resultado = danza.actualizar(); assertTrue("Prueba actualizar danza correctamente", resultado);</pre> | Se manda a llamar la función con los parámetros vacíos | El sistema obtiene todos los tipos de danza en la base de datos | El sistema muestra las danzas registradas en la bd y habilita campos para editar. | V |
| <pre>Danza danza = new Danza(); boolean resultado = danza.actualizar(); assertTrue("Prueba actualizar danza incorrectamente", resultado);</pre> | Se manda los datos para la función cuando no hay conexión o se manda incorrectamente. | El sistema no obtiene nada de la base de datos. | El sistema manda un mensaje de que no se pudo conectar a la base de datos o no existe el tipo de danza. | V |

| Función:EliminarDanza | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Danza danza = Danza.obtenerTodas().get(0); boolean resultado = danza.eliminar(); assertTrue("Prueba eliminar danza conrrectamente", resultado); | Se selecciona una danza y se manda a llamar a la función eliminar | El sistema borra el tipo de danza de la base de datos | El sistema manda un mensaje de que el tipo de danza fue eliminado correctamente | V |
| Danza danza = new Danza(); Danza danza = new Danza(); boolean resultado = danza.eliminar(); assertTrue("Prueba eliminarr danza inconrrectamente", resultado); | Se manda los datos para la función cuando no hay conexión o se manda incorrectamente. | El sistema no puede acceder a la base de datos. | El sistema manda un mensaje de que no se pudo conectar a la base de datos o no existe el tipo de danza. | V |

| Función:BuscarDanzaporNombre | | | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Danza danza = Danza.obtenerTodas().get(0); Danza danzaEncontrada = Danza.buscarPorTipoDanza(danza.getTipoDanza()); Assert.assertEquals("Prueba buscar danza por nombre de la danza",danzaEncontrada, danza); | Se manda la cadena con un nombre de tipo de danza para buscar en la base de datos | El sistema regresa el tipos de danza registradas en la bd, con ese nombre si la hay. | El sistema muestra en la tabla de danzas, todos los datos del tipo de danza seleccionado | V |

Clase Clase

| Función: ObtenerClasesPorGrupo | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>List<Clase> clases = Clase.obtenerClasesDelGrupo(17); boolean resultado = clases.size() > 0; Assert.assertTrue("Prueba obtener Clases por Id de grupo", resultado);</pre> | Se manda la id de la clase de danza para buscar en la base de datos | El sistema regresa la clase registrada en la bd, con ese nombre si la hay. | El sistema muestra en la tabla de clases, todos los datos del tipo de danza seleccionado | V |
| <pre>List<Clase> clases = Clase.obtenerClasesDelGrupo(-1); boolean resultado = clases.isEmpty(); Assert.assertTrue("Prueba No encontrar Clases por Id de grupo", resultado);</pre> | Se manda un id invalido de una clase de danza | El sistema no puede obtener los datos. | El sistema regresa un mensaje de advertencia de que la clase no existe | V |

| Función: CambiarDeGrupo | | | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>Clase clase = Clase.obtenerClasesDelGrupo(6).get(0); Grupo grupo = Grupo.listarGrupos().get(0); boolean resultado = clase.cambiarDeGrupo(grupo); Assert.assertTrue("Prueba cambiar grupo en clase", resultado);</pre> | Se manda la id de la clase de danza origen, además del id del alumno seleccionado y el grupo a donde se quiere cambiar el alumno | El sistema ejecuta la transacción reduciendo el numero de alumnos en el origen y copiando y aumentando el numero de alumnos en el destino | El sistema muestra en la tabla de clases, todos los datos de los alumnos así como el recién transferido. | V |
| <pre>Clase clase = Clase.obtenerClasesDelGrupo(6).get(0); boolean resultado = clase.cambiarDeGrupo(new Grupo()); Assert.assertTrue("Prueba cambiar grupo en clase incorrectamente", resultado);</pre> | Se manda un id de una clase que no existe o no se selecciona a un alumno | El sistema no puede acceder a la base de datos, dado que no existe la clase destino | El sistema manda un mensaje de que la clase destino no existe o que el alumno ya fue transferido | V |

Clase Egreso

| Función: CrearEgreso | | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|----------------------------------------------------|--------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Egreso egreso = new Egreso(); egreso.setMonto(0); egreso.setMotivo("Prueba "); egreso.setFecha(new Date(new GregorianCalendar().getTimeInMillis())); boolean resultado = egreso.crear(); assertTrue("Prueba registrar egreso correctamente", resultado); | Se ingresa un monto, un motivo y se elige la opción de crear egreso. | El sistema registra el egreso en la base de datos, | El egreso es visible en la base de datos. | V |
| Egreso egreso = new Egreso(); boolean resultado = egreso.crear(); assertTrue("Prueba registrar egreso incorrectamente", resultado); | Se selecciona la opción de crear un egreso sin mandar algún dato. | El sistema no puede registrar los datos. | El sistema regresa un mensaje de advertencia de que se tienen que llenar los campos. | V |

| Función: ObtenerTodosLosEgresos | | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|-----------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| List<Egreso> lista = Egreso.obtenerTodosLosEgresos(); boolean resultado = !lista.isEmpty(); assertTrue("Prueba obtener todos los egresos", resultado); | Se manda a llamar a la función que obtiene todos los egresos registrados en la base de datos, no es necesario algún parámetro. | El sistema regresa todos los egresos registrados en la bd. | El sistema muestra en la tabla de egresos, todos los datos del egreso | V |

Clase Promoción

| Función: CrearPromocion | | | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------|--------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>Promocion promocion = new Promocion(); promocion.setDescripcion("Promocion prueba"); promocion.setDescuento(0.4f); promocion.setTipo("Mensualidad"); boolean resultado = promocion.crear(); assertTrue("Prueba crear promocion correctamente", resultado);</pre> | Se ingresa una descripción, un descuento y se elige la mensualidad, después se elige crear promoción. | El sistema registra la promoción en la base de datos, | La promoción es visible y usable en la base de datos. | V |
| <pre>Promocion promocion = new Promocion(); boolean resultado = promocion.crear(); assertTrue("Prueba crear promocion correctamente", resultado);</pre> | Se selecciona la opción de crear una promoción sin mandar algún dato. | El sistema no puede registrar los datos. | El sistema regresa un mensaje de advertencia de que se tienen que llenar los campos. | V |

| Función: ActualizarPromocion | | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|--------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| <pre>Promocion promocion = Promocion.listaDePromociones().get(0); promocion.setDescripcion("Prueba actualizar promocion"); promocion.setDescuento(0.15f); promocion.setTipo("Inscripción"); boolean resultado = promocion.actualizar(); assertTrue("Prueba actualizar promocion correctamente", resultado);</pre> | Se ingresa una descripción, un descuento y se elige el tipo de la promoción después se elige actualizar promoción. | El sistema actualiza la promoción en la base de datos, | La promoción es visible en la base de datos. | V |
| <pre>Promocion promocion = new Promocion(); boolean resultado = promocion.actualizar(); assertTrue("Prueba actualizar promocion incorrectamente", resultado);</pre> | Se selecciona la opción de crear una promoción sin mandar algún dato. | El sistema no puede actualizar los datos. | El sistema regresa un mensaje de advertencia de que se tienen que llenar los campos. | V |

| Función: EliminarPromocion | | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Promocion promocion = Promocion.listaDePromociones().get(0); boolean resultado = promocion.eliminar(); assertTrue("Prueba eliminar promocion correctamente", resultado); | Se selecciona una promoción y se elige eliminar | El sistema borra la promoción de la base de datos | La promoción desaparece de las disponibles | V |
| Promocion promocion = new Promocion(); boolean resultado = promocion.eliminar(); assertTrue("Prueba eliminar promocion correctamente", resultado); | Se selecciona la opción de eliminar sin elegir un dato. | El sistema no puede acceder a los datos. | El sistema regresa un mensaje de advertencia de que se tienen seleccionar una promoción | V |

| Función: ObtenerTodasLasPromociones | | | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|-------------------------------------------------------------------|---------------------------------------------------------|-----------|
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| List<Promocion> lista = Promocion.listaDePromociones(); boolean resultado = !lista.isEmpty(); assertTrue("Prueba obtener todos los resultados", resultado); | Se llama a la función y se obtienen los datos de promociones registradas | El sistema obtiene los datos de todas las promociones disponibles | Las promociones son visibles en la tabla de promociones | V |