

Documento de Primera Entrega

Luis Fernando Gomez Alejandre Francisco Gerardo Mares Solano



30 de marzo de 2017

Universidad Veracruzana

Contenido

[Descripción de los casos de uso implementados 2](#_Toc478652787)

[CRUD Alumnos 2](#_Toc478652788)

[CRUD Grupos 4](#_Toc478652789)

[CRUD Promociones 6](#_Toc478652790)

[CRUD Colaboradores 8](#_Toc478652791)

[Diagramas de robustez de cada CU 10](#_Toc478652792)

[Diagrama de dominio 13](#_Toc478652793)

[Diagrama de clases 13](#_Toc478652794)

[Diagramas de secuencia de cada CU 14](#_Toc478652795)

[Modelo relacional 19](#_Toc478652796)

[Plan de pruebas 19](#_Toc478652797)

# 

# Diagrama de Casos de Uso



# Descripción de los casos de uso implementados

### CRUD Alumnos

Tabla 1 Tabla de caso de uso CRUD Alumnos

|  |  |
| --- | --- |
| **ID:** | CU\_01 |
| **Nombre:** | CRUD Alumnos |
| **Autor(es):** | Luis Fernando Gomez Alejandre |
| **Fecha de creación:** | 14/02/2017 |
| **Fecha de actualización:** | 10/05/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario quiere realizar alguna acción sobre los alumnos, ya sea registrar uno nuevo, editarlo o darlo de baja. |
| **Precondiciones:** |  |
| **Flujo Normal:** | **Flujo normal**  1. El Usuario Selecciona la opción de Administrar Alumnos  2. El sistema muestra las 3 opciones del CRUD Alumnos  3. El Usuario selecciona Agregar Alumno  **Alternate: 3a. Editar Alumno Alternate: 3b. Dar de baja Alumno**  4. El sistema muestra [Invokes: Agregar Alumno]  5. El Usuario rellena el formulario  6. El sistema muestra la pantalla de confirmación  7. El Usuario guarda el registro del Alumno  **Exception: 7a. El sistema no puede hacer conexión a base de datos**  8. El sistema muestra pantalla de operación exitosa |
| **Flujos Alternos:** | **Alternate. Dar de baja Alumno**  1. El Usuario selecciona Dar de baja a Alumnos  2. El sistema muestra [Invokes: Dar Baja Alumno]  3. El Usuario busca al Alumno que será dado de baja  **4. Regreso al Flujo Normal Punto 6**  **Alternate. Editar Alumno**  1. El Usuario selecciona Editar Alumno  2. El sistema muestra [Invokes: Editar Alumno]  3. El Usuario busca al Alumno  4. El sistema muestra información acerca del Alumno  5. El Usuario modifica la información existente del Alumno |
| **Excepciones:** | **Excepción: 7a. El sistema no puede hacer conexión a base de datos**  1. El sistema muestra un mensaje de error donde dice que no se puedo conectar a la base de datos. |
| **Poscondiciones:** | **POST01-**El alumno puede ser editado  **POST02-**El alumno puede ser dado de baja  **POST03-**El alumno puede ser inscrito a un grupo |
| **Prioridad:** | Indispensable |

### CRUD Colaboradores

Tabla 2 Tabla de caso de uso CRUD Colaboradores

|  |  |
| --- | --- |
| **ID:** | CU\_02 |
| **Nombre:** | CRUD Colaboradores |
| **Autor(es):** | Luis Fernando Gomez Alejandre |
| **Fecha de creación:** | 14/02/2017 |
| **Fecha de actualización:** | 10/05/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario quiere realizar alguna acción sobre los Colaboradores, ya sea registrar uno nuevo, editarlo o darlo de baja. |
| **Precondiciones:** |  |
| **Flujo Normal:** | **Flujo normal**  1. El Usuario Selecciona la opción de Administrar Colaboradores  2. El sistema muestra las 3 opciones del CRUD Colaboradores  3. El Usuario selecciona Agregar Colaboradores  **Alternate: 3a. Editar Colaborador** **Alternate: 3b. Dar de baja Colaborador**  4.El sistema muestra [Invokes: Agregar Colaborador]  5. El Usuario rellena el formulario  6. El sistema muestra la pantalla de confirmación  7. El Usuario guarda el registro del Colaborador  **Exception: 7a. El sistema no puede hacer conexión a base de datos**  8. El sistema muestra pantalla de operación exitosa |
| **Flujos Alternos:** | **Alternate. Dar de baja Colaborador**  1. El Usuario selecciona Dar de baja a Colaborador  2. El sistema muestra [Invokes: Dar Baja Colaborador]  3. El Usuario busca al Colaborador que será dado de baja  **4. Regreso al Flujo Normal Punto 6**  **Alternate. Editar Colaborador**  1. El Usuario selecciona Editar Colaborador  2. El sistema muestra [Invokes: Editar Colaborador]  3. El Usuario busca al Colaborador  4. El sistema muestra información acerca del Colaborador  5. El Usuario modifica la información existente del Colaborador |
| **Excepciones:** | **Excepción: 7a. El sistema no puede hacer conexión a base de datos**  1. El sistema muestra un mensaje de error donde dice que no se puedo conectar a la base de datos. |
| **Poscondiciones:** | **POST01-**El Colaborador puede ser editado  **POST02-**El Colaborador puede ser dado de baja  **POST03-**El Colaborador puede ser inscrito a un grupo |
| **Prioridad:** | Indispensable |

### CRUD Grupos

Tabla 3 Tabla de caso de uso CRUD Grupos

|  |  |
| --- | --- |
| **ID:** | CU\_03 |
| **Nombre:** | CRUD Grupos |
| **Autor(es):** | Luis Fernando Gomez Alejandre |
| **Fecha de creación:** | 15/02/2017 |
| **Fecha de actualización:** | 16/02/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario quiere realizar alguna acción sobre los grupos, ya sea registrar uno nuevo, editarlo o darlo de baja. |
| **Precondiciones:** | Debe haber registros en la base de datos |
| **Flujo Normal:** | 1. El Usuario ingresa al apartado de control de grupos  2. El sistema muestra los grupos disponibles  3. El Usuario ingresa a uno de los grupos  **Alternate: 3a. Eliminar Grupo Alternate: 3b. Nuevo Grupo**  4. El sistema muestra los alumnos registrados en el grupo  5. El Usuario agrega al nuevo alumno al curso  6. El sistema pide confirmación de las acciones  7. El Usuario confirma las respuestas |
| **Flujos Alternos:** | **Nuevo Grupo**  1. El Usuario ingresa a nuevo grupo  2. El sistema despliega una ventana  3. El Usuario ingresa los datos del nuevo curso  **4. Retorno a Punto 6 del Flujo Normal**  **Eliminar Grupo**  1. El Usuario ingresa al curso que desea eliminar  2. El sistema despliega el grupo con los datos  3. El Usuario elimina el grupo  **4. Retorno a Punto 6 del Flujo Normal** |
| **Excepciones:** | **No hay conexión en la base de datos**  1. El sistema no puede hallar la base de datos o no hay conexión |
| **Poscondiciones:** | **POST01-**El Grupo puede ser editado  **POST02-**El Grupo puede ser dado de baja  **POST03-**El Grupo puede ser inscrito a un grupo |
| **Prioridad:** | Indispensable |

### CRUD Promociones

Tabla 4 Tabla de caso de uso CRUD Promociones

|  |  |
| --- | --- |
| **ID:** | CU\_04 |
| **Nombre:** | CRUD Promociones |
| **Autor(es):** | Luis Fernando Gomez Alejandre |
| **Fecha de creación:** | 15/02/2017 |
| **Fecha de actualización:** | 10/04/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario quiere realizar alguna acción sobre las promociones, ya sea registrar uno nuevo 0 editarlo. |
| **Precondiciones:** | Deben existir registros en la base de datos |
| **Flujo Normal:** | 1. El Usuario entra al apartado de promociones  2. El sistema despliega las promociones existentes  3. El Usuario agrega una nueva promoción  **Alterno: 3a. Eliminar Promoción**  4. El sistema despliega **Agregar Promoción**  5. El Usuario rellena los campos de la promoción  **Excepción: 5a. No hay conexión a la base de datos**  6. El sistema confirma las operaciones realizadas  7. El Usuario acepta los cambios |
| **Flujos Alternos:** | **Eliminar Promoción**  1. El Usuario selecciona la promoción a eliminar  2. El sistema muestra la confirmación de la eliminación  **3. Retorno al Flujo Normal punto 6** |
| **Excepciones:** | **No hay conexión a la base de datos**  1. El sistema no puede continuar por que no hay conexión a la base de datos |
| **Poscondiciones:** | **POST01-**La promoción puede ser dado de baja  **POST02-**La promoción puede ser inscrito a un grupo |
| **Prioridad:** | Indispensable |

Registrar Asistencia

Tabla 5 Tabla de caso de uso Registrar Asistencia

|  |  |
| --- | --- |
| **ID:** | CU\_05 |
| **Nombre:** | Registrar asistencia |
| **Autor(es):** | Francisco Gerardo Mares Solano |
| **Fecha de creación:** | 17/04/2017 |
| **Fecha de actualización:** | 17/04/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el Usuario decide registrar la asistencia de los alumnos, el sistema le proporciona una lista de grupos disponibles para tal acción y el usuario les <<pasa lista>>. |
| **Precondiciones:** | **PRE01-**Alumnos registrados en la BD y activos.  **PRE02-**Colaboradores registrados en la BD y activos.  **PRE03-**Grupos registrados en la BD y activos.  **PRE04-**El día actual debe ser el día en que el grupo tendrá clases. |
| **Flujo Normal:** | 1. El Usuario ingresa al apartado Registrar Asistencia  2. El sistema muestra los grupos a los cuales se le puede pasar lista  **Exception: 2a. Error en la BD**  3. El Usuario selecciona un grupo  4. El sistema muestra una lista con todos los alumnos inscritos en el grupo y una lista vacía a donde podrá pasar los alumnos que asistieron  5. El Usuario selecciona al alumno con asistencia y guarda la lista  **Alternate: 5a. El usuario se arrepiente**  6. El sistema pide confirmación del guardado de lista Uses:  7. El Usuario confirma que quiere guardar la lista Uses:  8. El sistema le informa que la lista fue guardada |
| **Flujos Alternos:** | **Alternate. El usuario se arrepiente**  1. El usuario prefiere no guardar la lista y de clic en cancelar |
| **Excepciones:** | **Exception. Error en la BD**  1. El sistema tiene un error al comunicarse con la BD e informa al usuario Uses:  2. El usuario indica que entendió |
| **Poscondiciones:** |  |
| **Prioridad:** | Deseable |

Configuración General del Sistema

Tabla 6 Tabla de caso de uso Configuración General del Sistema

|  |  |
| --- | --- |
| **ID:** | CU\_06 |
| **Nombre:** | Configuración general del sistema |
| **Autor(es):** | Francisco Gerardo Mares Solano |
| **Fecha de creación:** | 17/04/2017 |
| **Fecha de actualización:** | 17/04/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario decide modificar los valores de configuración general del sistema, el sistema lo dirige a la configuración y una vez hechos los cambios pertinentes, las modificaciones son validadas y guardadas |
| **Precondiciones:** |  |
| **Flujo Normal:** | 1. El Usuario se dirige a la configuración general del sistema  2. El sistema le muestra los campos de configuración general del sistema  3. El usuario modifica los campos y da clic en guardar  **Alternate: 3a. Cancelación de cambios**  4. El sistema le informa que los cambios han sido debidamente guardados Uses:  **Exception: 4a. Error en la BD** |
| **Flujos Alternos:** | **Alternate. Cancelación de cambios**  1. El usuario decide cancelar los cambios en la configuración general del sistema |
| **Excepciones:** | **Exception. Error en la BD**  1. El sistema le informa que ha habido un error al guardar los cambios en la BD |
| **Poscondiciones:** |  |
| **Prioridad:** | Indispensable |

Generar Reporte Mensual

Tabla 7 Tabla de caso de uso Generar Reporte Mensual

|  |  |
| --- | --- |
| **ID:** | CU\_07 |
| **Nombre:** | Generar Reporte Mensual |
| **Autor(es):** | Francisco Gerardo Mares Solano |
| **Fecha de creación:** | 17/04/2017 |
| **Fecha de actualización:** | 17/04/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario desea generar un reporte mensual de ingresos y egresos. El sistema únicamente le pide la ruta dónde lo guardará, así como el nombre y con eso, genera el reporte el sistema. |
| **Precondiciones:** |  |
| **Flujo Normal:** | 1. El Usuario ingresa al apartado de reportes  2. El sistema le pide la ruta donde guardar el reporte y el nombre  3. El Usuario elige la ruta, el nombre y genera el reporte  **Exception: 3a. No Se Pudo Generar el Reporte**  4. El sistema informa que el reporte ha sido generado  **Alternate: 4a. Reporte cancelado** |
| **Flujos Alternos:** | **Alternate. Reporte cancelado**  1. El usuario decide cancelar la generación del reporte Uses:  2. El sistema le informa que el reporte ha sido cancelado |
| **Excepciones:** | **Exception. No Se Pudo Generar el Reporte**  1. El sistema informa que el reporte no se pudo generar |
| **Poscondiciones:** |  |
| **Prioridad:** | deseable |

Registrar Egreso

Tabla 8 Tabla de caso de uso Registrar Egreso

|  |  |
| --- | --- |
| **ID:** | CU\_08 |
| **Nombre:** | Registrar Egreso |
| **Autor(es):** | Francisco Gerardo Mares Solano |
| **Fecha de creación:** | 17/04/2017 |
| **Fecha de actualización:** | 17/04/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario quiere registrar un egreso. El sistema le proporciona formularios de egresos para pago a colaboradores y otro para gastos, una vez que el usuario lo rellena y decide registrar el egreso, el sistema agrega el registro. |
| **Precondiciones:** |  |
| **Flujo Normal:** | 1. El Usuario ingresa al apartador de Egreso  **Alternate: 1a. Registrar Gastos**  2. El sistema despliega un menú con [Invokes: Registrar Pago a Colaborador] inicialmente  3. El Usuario seleciona el tipo de egreso y rellena los campos solicitados y selecciona registrar el egreso  **Alternate: 3a. Cancelar acción**  4. El sistema pide confirmación del registro de gastos  5. El Usuario Confirma los datos  6. El sistema regresa al usuario a la ventana de egresos  Exception: 6ª. El Reporte No Se Pudo Generar |
| **Flujos Alternos:** | **Alternate. Registrar Gastos**  1. El sistema muestra [Invokes: Registrar Gastos]  2. El Usuario establece el monto y el concepto  3. El sistema muestra la confirmación y regresa al flujo normal 6  **Alternate. Cancelar acción**  1. El usuario decide no registrar ningún egreso y da clic en cancelar |
| **Excepciones:** | **Exception. El reporte no se pudo generar**  1. El sistema le informa al usuario que el reporte no pudo generarse |
| **Poscondiciones:** | **POST01**- Generar un recibo de egreso |
| **Prioridad:** | deseable |

Registrar Ingreso

Tabla 9 Tabla de caso de uso Registrar Ingreso

|  |  |
| --- | --- |
| **ID:** | CU\_09 |
| **Nombre:** | Registrar Ingreso |
| **Autor(es):** | Francisco Gerardo Mares Solano |
| **Fecha de creación:** | 17/04/2017 |
| **Fecha de actualización:** | 17/04/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario quiere registrar un ingreso. El sistema le proporciona formularios de ingreso para el pago de alumnos por mensualidad o anualidad, una vez que el usuario lo rellena y decide registrar el ingreso, el sistema agrega el registro |
| **Precondiciones:** |  |
| **Flujo Normal:** | 1. El Usuario ingresa al apartado de egresos  2. El sistema muestra opciones de pago a los colaboradores  **Exception: 2a. Error Base de datos**  3. El Usuario selecciona al colaborador a pagar  **Alternate: 3a. Cancelar acción**  4. El sistema muestra notificación de pago  5. El Usuario [Invokes: Generar Recibo] |
| **Flujos Alternos:** | **Alternate. Cancelar acción**  1. El usuario decide no registrar ningún ingreso y da clic en cancelar |
| **Excepciones:** | **Exception. Error Base de datos**  1. El sistema no puede mostrar a los colaboradores, error al conectar a la base de datos Uses:  2. El Usuario sale del sistema para reintenar despues |
| **Poscondiciones:** | **POST01**- Generar un recibo de ingreso |
| **Prioridad:** | deseable |

Generar Recibo

Tabla 10 Tabla de caso de uso Generar Recibo

|  |  |
| --- | --- |
| **ID:** | CU\_10 |
| **Nombre:** | Generar Recibo |
| **Autor(es):** | Francisco Gerardo Mares Solano |
| **Fecha de creación:** | 15/02/2017 |
| **Fecha de actualización:** | 10/04/2017 |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia después de que el usuario registró el pago de un alumno o a algún colaborador. Únicamente genera un recibo en la ubicación y con el nombre elegido |
| **Precondiciones:** | **PRE01-**Una acción de ingreso o egreso de capital debe realizarse |
| **Flujo Normal:** | 1. El usuario aceptó registrar el movimiento de capital  2. El sistema le pide al usuario la ruta donde se guardará el archivo  3. El usuario elige la ruta del archivo  **Alternate: 3a. Cancelación de recibo**  4. El sistema informa que el recibo fue generado correctamente  **Exception: 4a. No Puede Generar Recibo** |
| **Flujos Alternos:** | **Alternate. Cancelación de recibo**  1. El usuario cancela la generación del recibo y registro de pago  2. El sistema informa que el movimiento ha sido cancelado |
| **Excepciones:** | **Exception. No Puede Generar Recibo**  1. El sistema informa que no fue posible generar el recibo |
| **Poscondiciones:** |  |
| **Prioridad:** | deseable |

Ver Asistencia

Tabla 10 Tabla de caso de uso Generar Recibo

|  |  |
| --- | --- |
| **ID:** | CU\_11 |
| **Nombre:** | Generar Recibo |
| **Autor(es):** | Francisco Gerardo Mares Solano |
| **Fecha de creación:** | 08/06/2017 |
| **Fecha de actualización:** |  |
| **Actor(es):** | Usuario |
| **Descripción:** | Este caso de uso inicia cuando el usuario quiere ver quienes asistieron a clases en el día |
| **Precondiciones:** | **PRE01-**Debe haber alumnos registrados  **PRE02-**Debe haber grupos registrados  **PRE03-**Debe haber alumnos registrados  **PRE04**-Se debe haber pasado lista a al menos un grupo |
| **Flujo Normal:** | 1. El usuario ingresa a la sección “Ver lista”  2. El sistema le pide muestra al usuario una lista de grupos a los cuales se les pasó lista en el día  **Alternate: 2a. No hay grupos**  3. El usuario elige un grupo  4. El sistema lleva al usuario a otra pantalla dónde se ve cuales alumnos han asistido  **Exception: 4a. No hay conexión a la BD** |
| **Flujos Alternos:** | **Alternate. No hay grupos**  1. El sistema le muestra al usuario un aviso dónde se le informa que no hay datos a visualizar  2. El usuario acepta que entendió el mensaje |
| **Excepciones:** | **Exception. No hay conexión a la BD**  1. El sistema informa que hay problemas con la conexión a la BD |
| **Poscondiciones:** |  |
| **Prioridad:** | deseable |

# Diagramas de robustez de cada CU

CRUD Alumnos



Figura 1 Diagrama de robustez CRUD AlumnosCRUD Colaboradores



Figura 2 Diagrama de robustez CRUD Colaboradores

CRUD Colaboradores

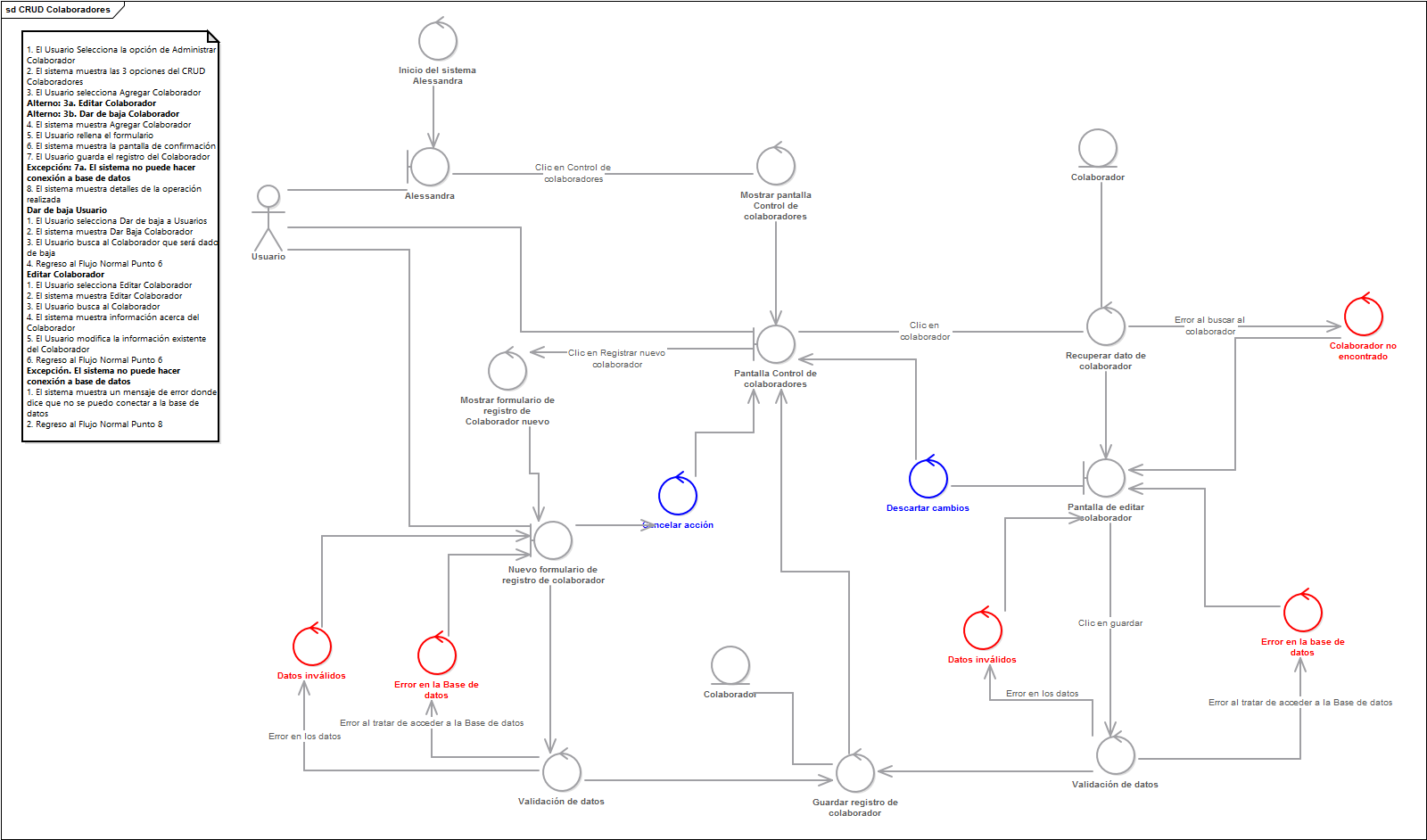


Figura 3 Diagrama de robustez CRUD Colaboradores

CRUD Grupos



Figura 4 Diagrama de robustez CRUD Grupos

CRUD Promociones



Figura 5 Diagrama de robustez CRUD Promociones

Registrar Asistencia



Figura 6 Diagrama de robustez Registrar asistencia

Configuración General Del Sistema



Figura 7 Diagrama de robustez Configuración General del Sistema

Generar Recibo



Figura 8 Diagrama de robustez Generar Recibo

Generar Reporte Mensual



Figura 9 Diagrama de robustez Generar Reporte Mensual

Registrar Egreso

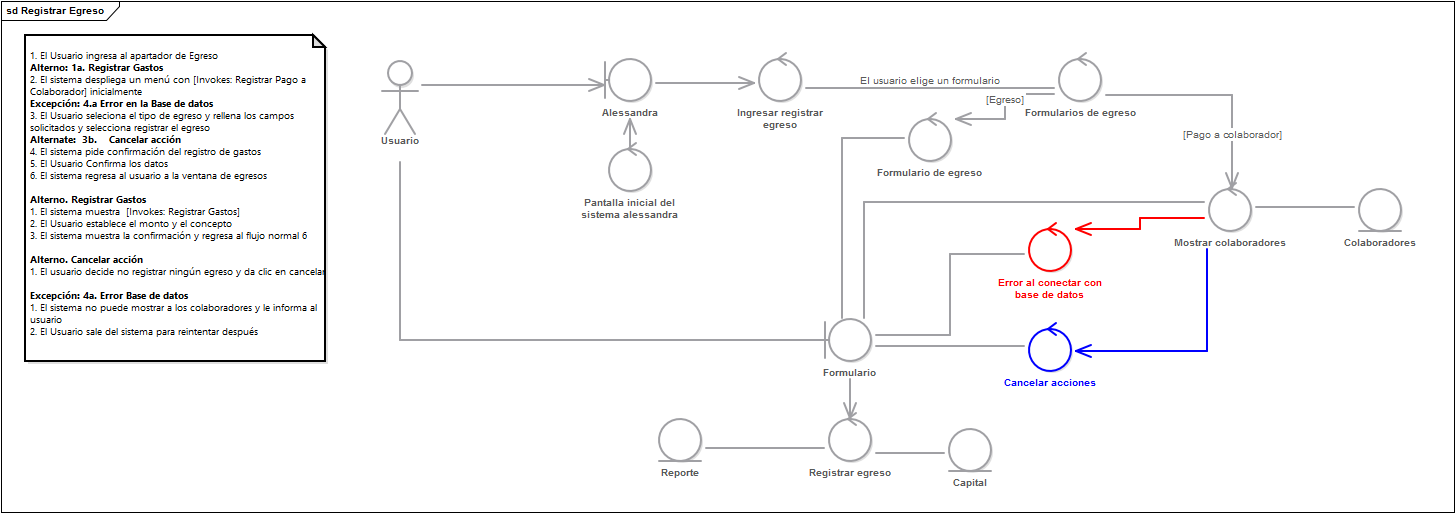


Ilustración 1 Diagrama de robustez Registrar Egreso

Registrar Ingreso

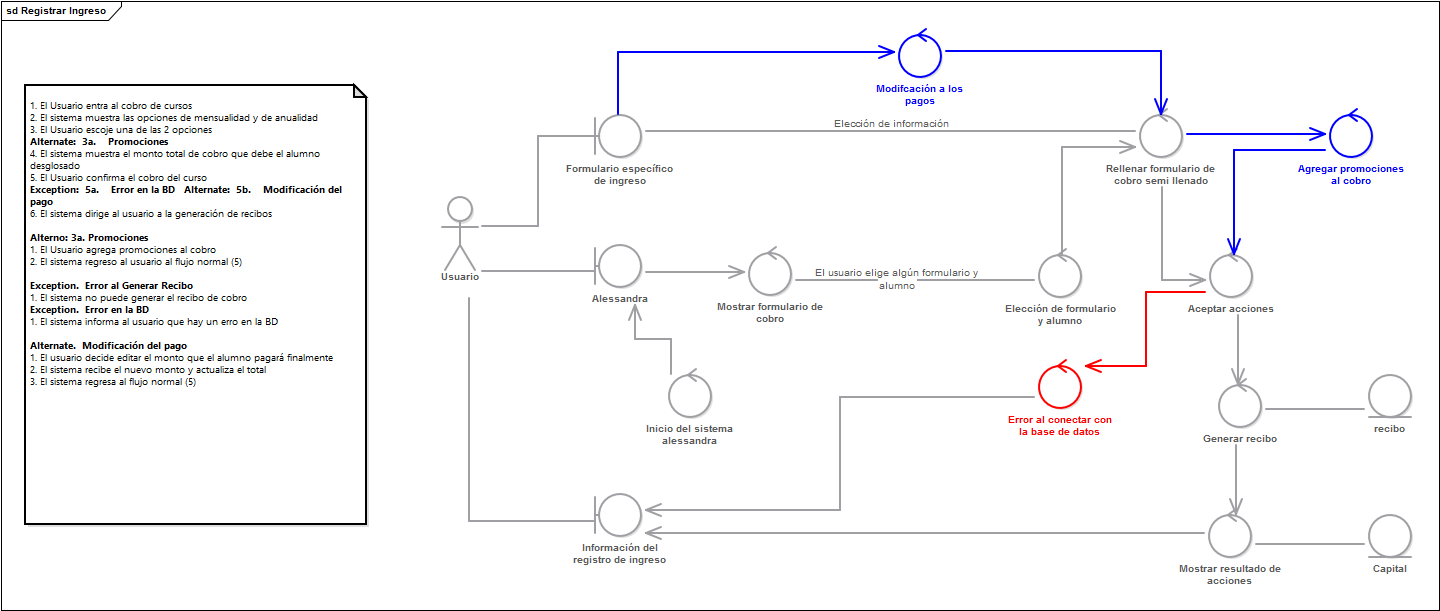


Ilustración 2 Diagrama de robustez Registrar Ingreso

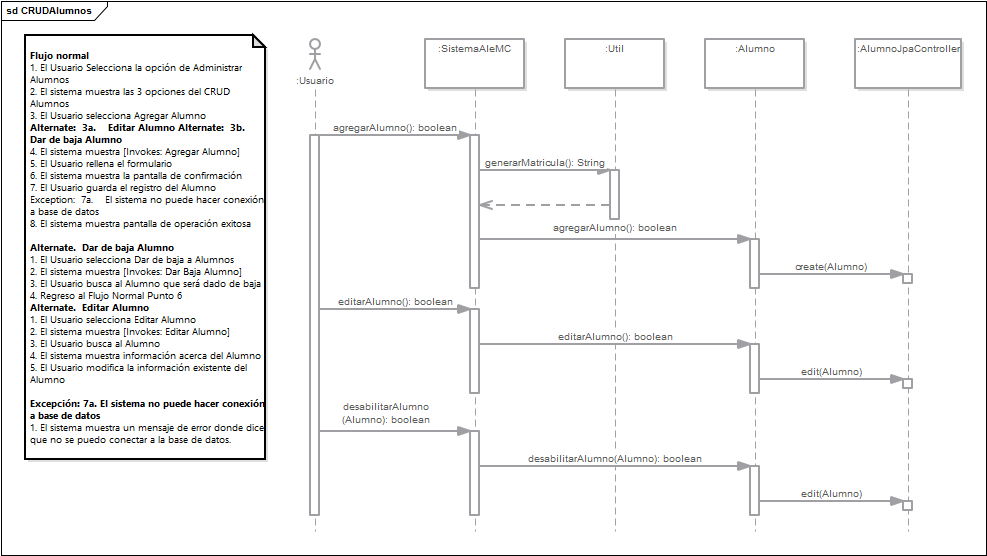
### Ver Asistencia



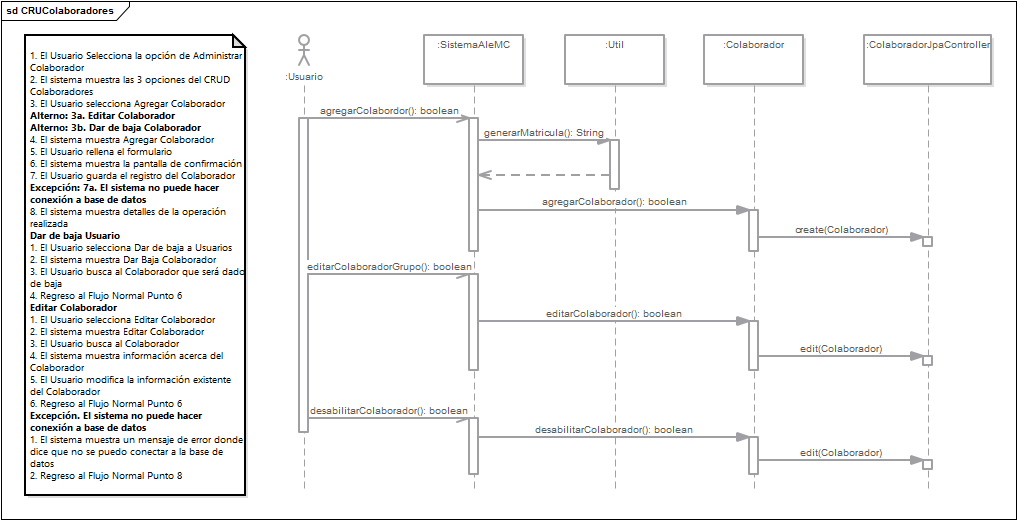
Ilustración 3 Diagrama de robustez Ver Asistencia

# Diagramas de secuencia de cada CU

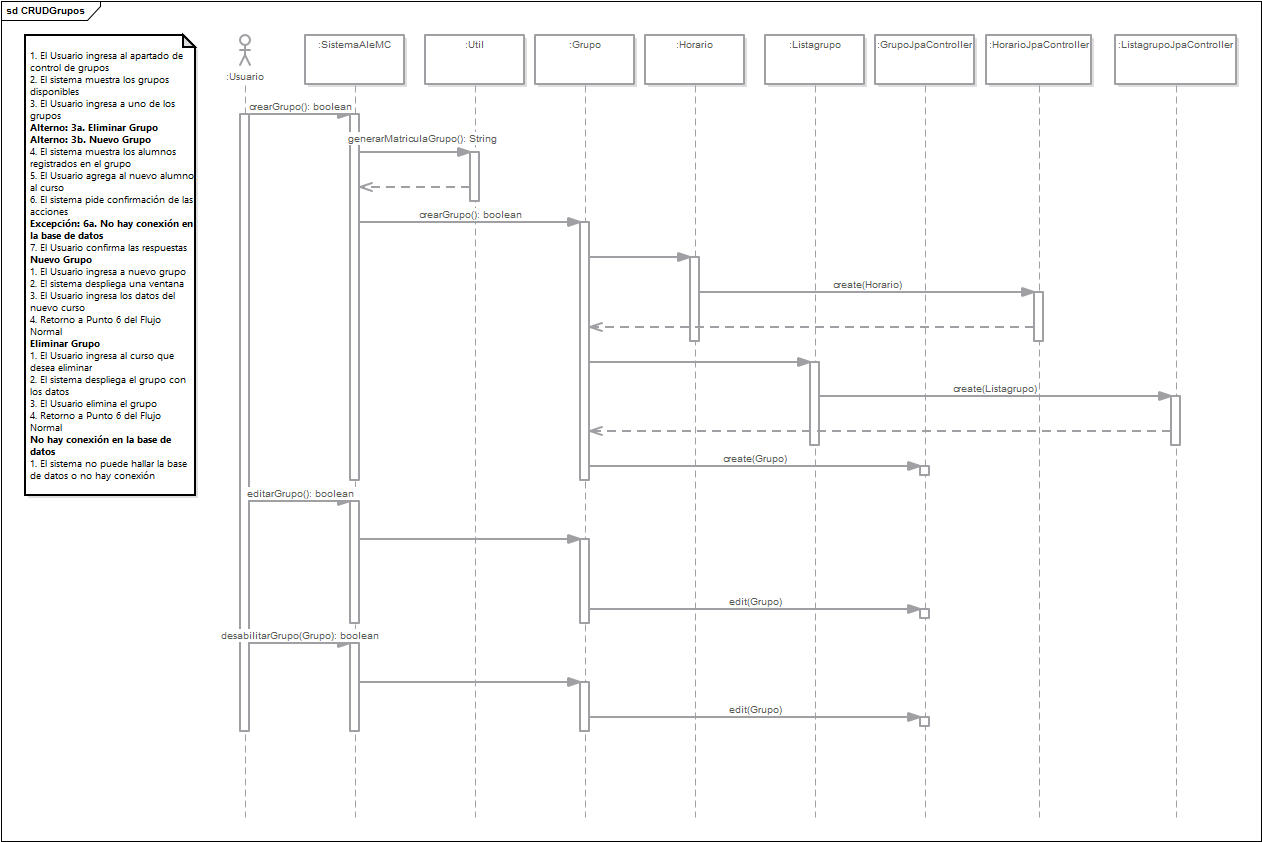
CRUD Alumnos



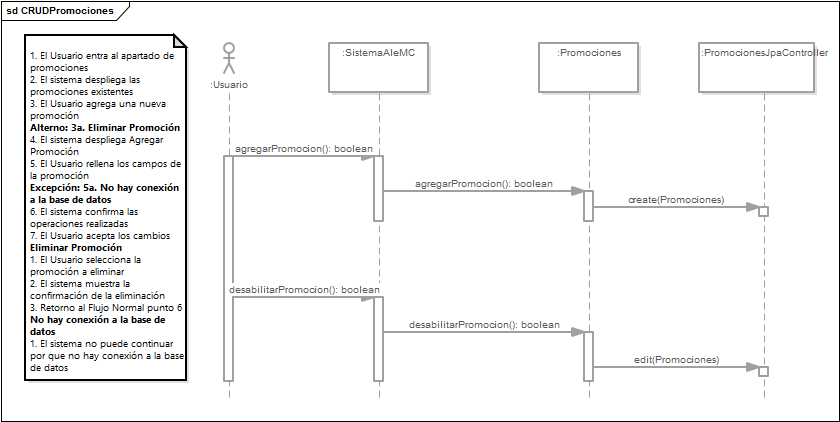
CRUD Colaboradores



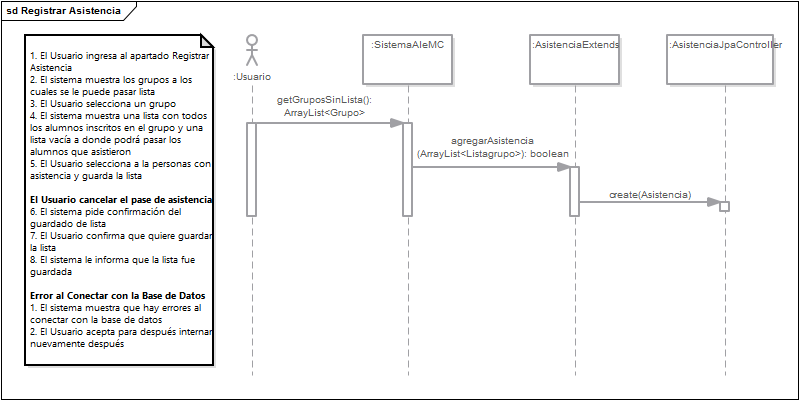
CRUS Grupos



CRUD Promociones

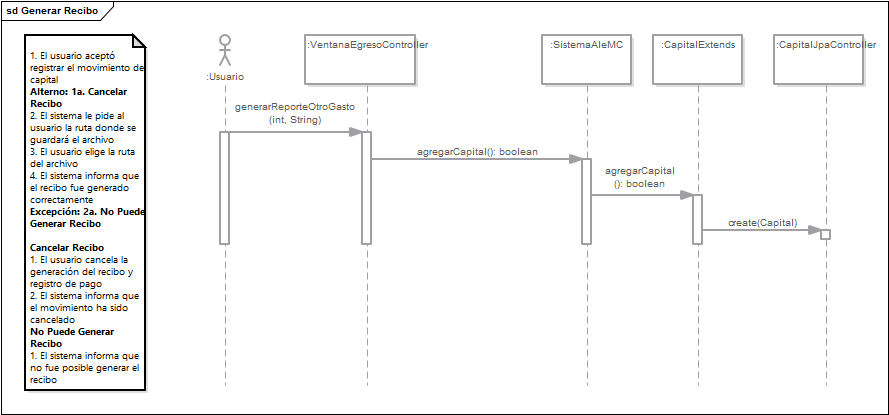


Registrar Asistencia

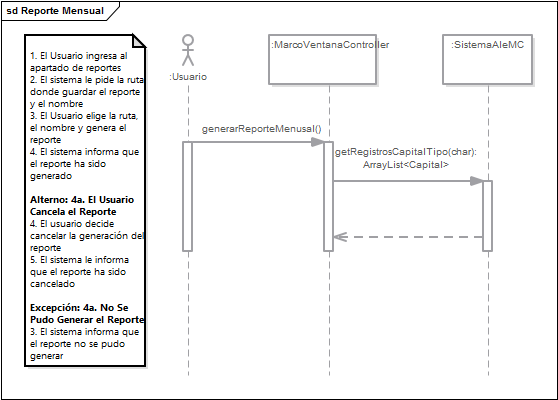


Configuración General del Sistema

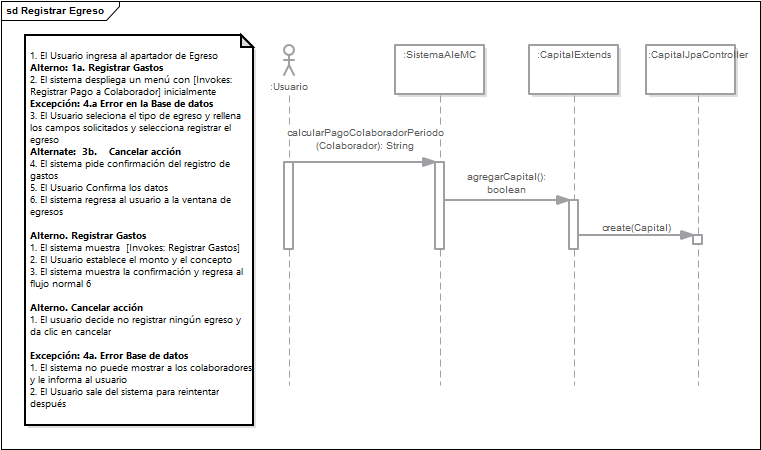
Generar Recibo



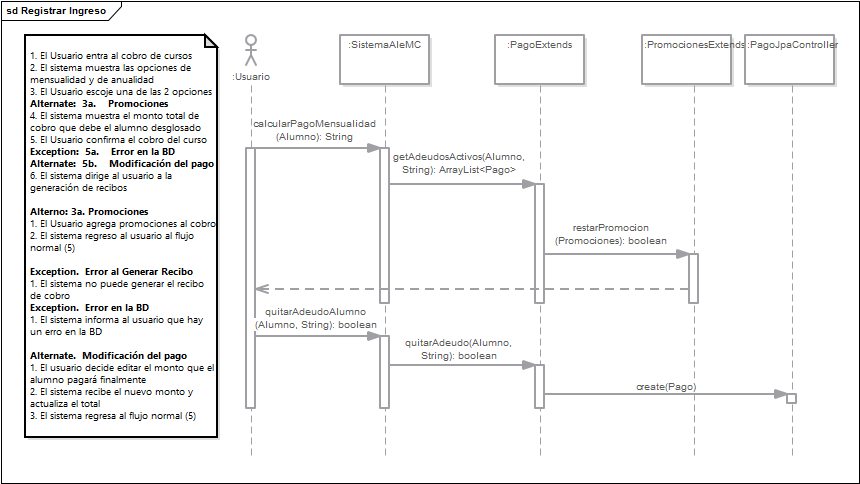
Generar Reporte Mensual



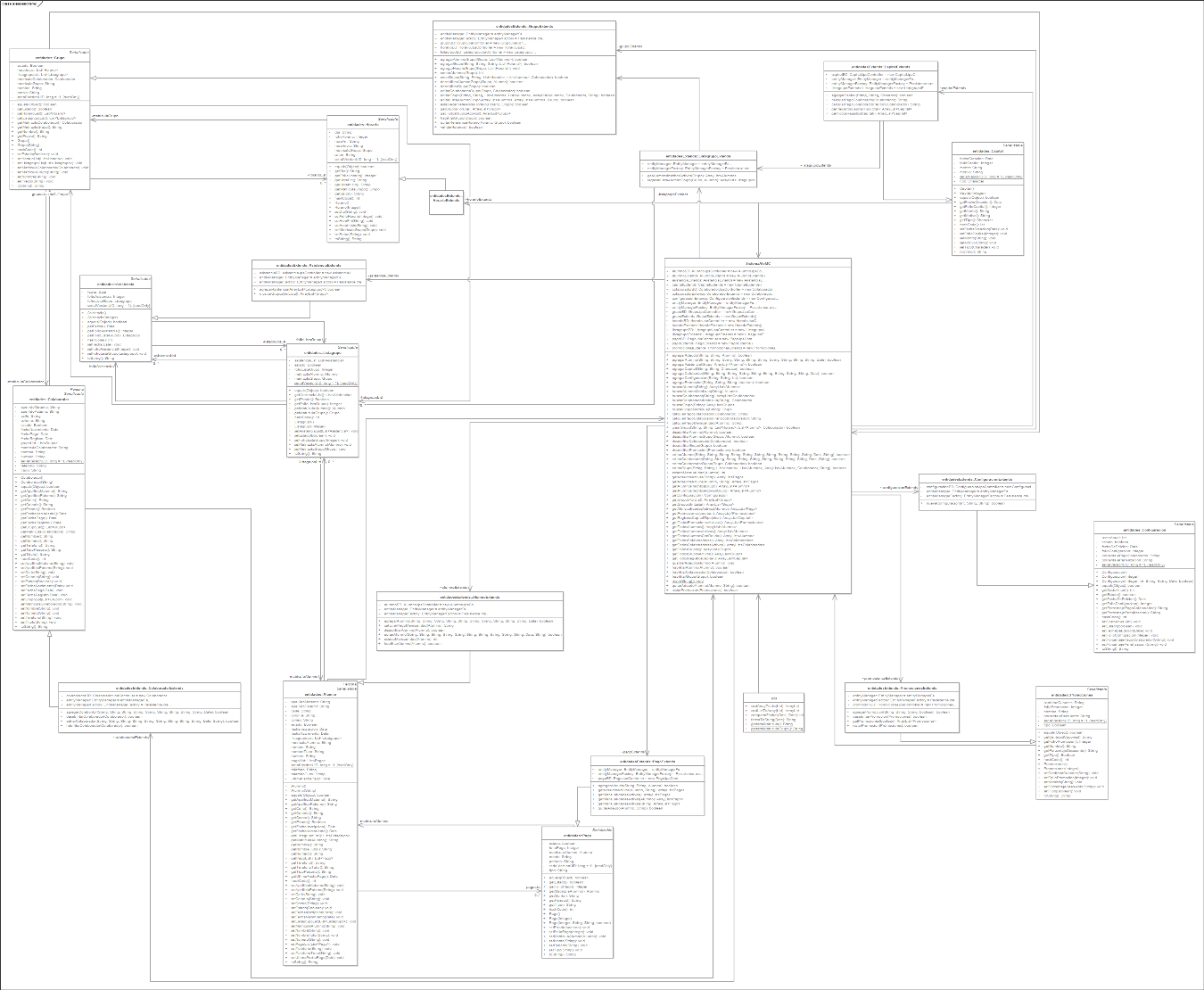
Registrar Egreso



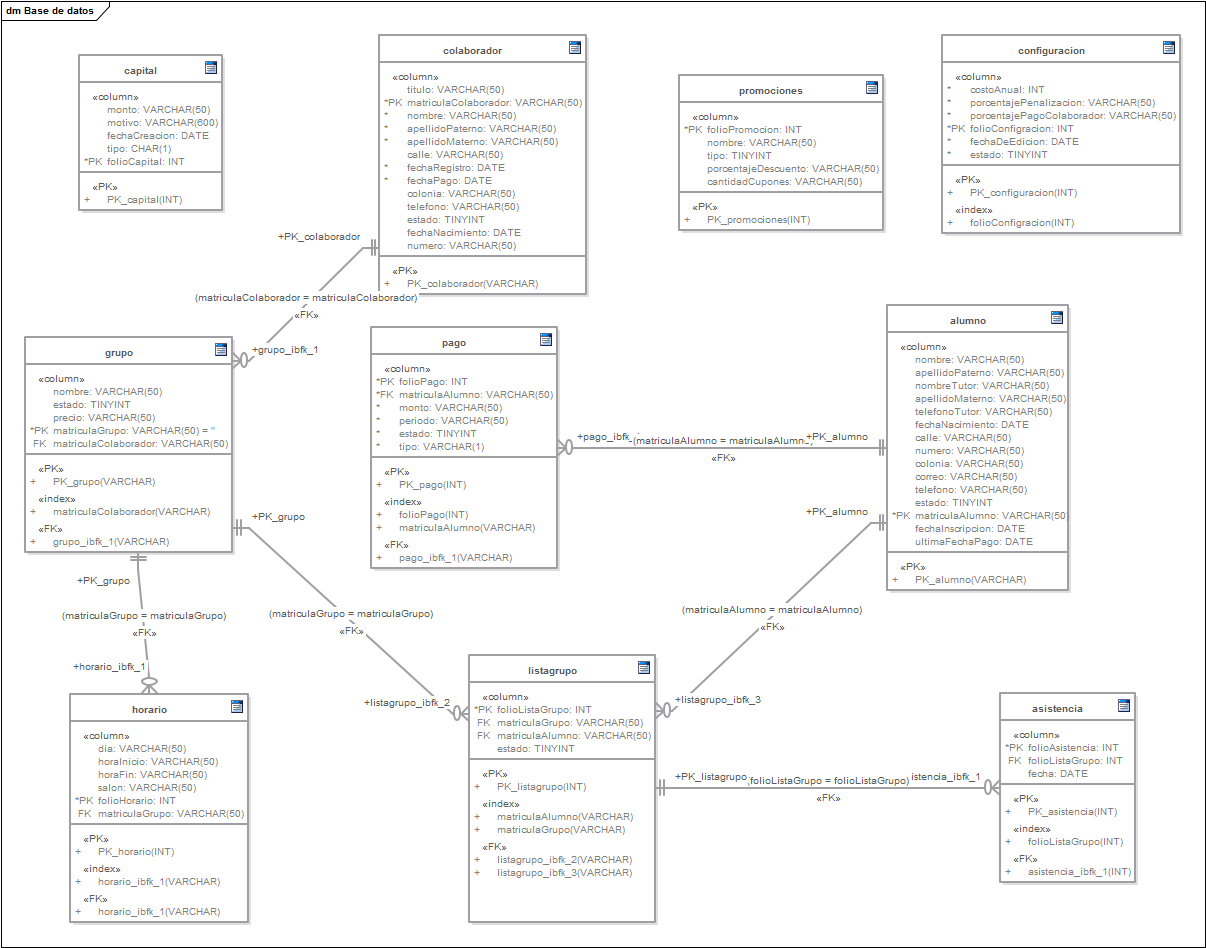
Registrar Ingreso



# Diagrama de clases



# Modelo relacional



# Plan de pruebas

## Clase Alumno

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Función: EditarAlumno | | | | |
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| "Alejandre", "Gomez", "Buena Vista", "Buena Vista", "gomez.alejandre@gmail.com", "Luis Fernando", "Alma Rosa", "6", "2282229579", "2281114099", fechaNacimiento, matriculaAlumno | El alumno existen en la BD | 1 | 1 | V |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Función: DesabilitarAlumno | | | | |
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Alumno válido |  | 1 | 1 | V |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Función: habilitarAlumno | | | | |
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| Alumno deshabilitado | Alumno deshabilitado | 1 | 1 | V |

## Clase Maestro

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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("");  Foto(null);  Genero(true);  Telefono("");  TelefonoMovil(""); | Los campos vacíos para registrar un maestro | Mensaje de advertencia | El sistema despliega un mensaje de advertencia, de que los campos están vacíos | V |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |
| 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); | 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 |
| setNivel("");  setSalon("");  grupo.setTipoDeDanza("");  Maestro maestro = Maestro.obtenerMaestroPorApellido("").get(0);  grupo.setIdMaestro(maestro);  ArrayList<Horario> horarios = new ArrayList<>();  boolean resultado = grupo.crear(horarios); | El grupo no existe en la base de datos, pero los campos están vacios. | El sistema manda una advertencia. | El sistema manda una advertencia de que los campos están vacios y deben verificarse. | V |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Función: ActualizarGrupo | | | | |
| Entradas | Condiciones de Entrada | Salida | Salida Esperada | Resultado |
| 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); | 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 |
| setNivel("");  setSalon("");  grupo.setTipoDeDanza("");  Maestro maestro = Maestro.obtenerMaestroPorApellido("").get(0);  grupo.setIdMaestro(maestro);  ArrayList<Horario> horarios = new ArrayList<>();  boolean resultado = grupo.crear(horarios); | 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: 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 |
| horario.setDia("Lunes"); horario.setHora("12:00-14:00");  Grupo grupo = Grupo.listarGrupos().get(0);  horario.setIdGrupo(grupo);  boolean resultado = horario.crear(); | 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 |
| Horario horario = new Horario();  horario.crear(); | 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 solictado, 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 |

No existe plan de pruebas