



Tecnológico de Monterrey

Construcción de Software y Toma de Decisiones

Avance 2: Análisis y diseño de la solución

Ayetza Yunnuen Infante García A01709011

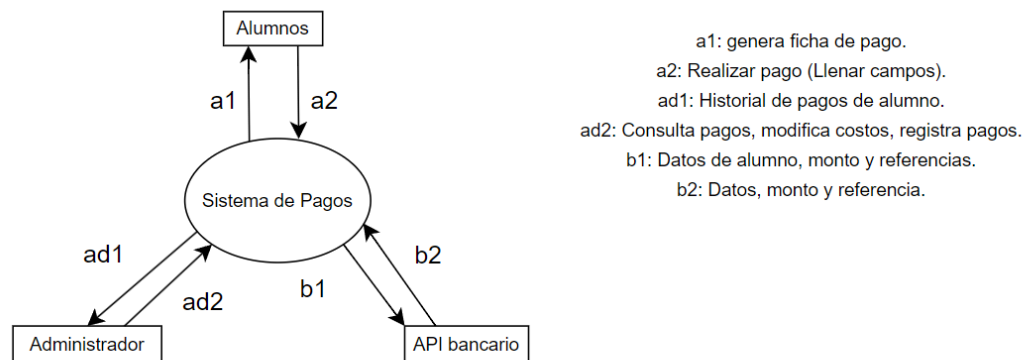
Maria Guadalupe Soria Velazquez A01710797

Miguel Angel Becerra Ayala A01710076



8 de abril del 2024

Diagrama de contexto



Requisitos funcionales

1. Usuario:

- Usuario registrado inicia sesión.
- Usuario registrado consulta el historial de pagos.
- Usuario consulta estado de cuenta de colegiatura.
- Usuario consulta estado de cuenta de otros servicios.

2. Alumno:

- Alumno realiza pago en línea.

3. Administrador:

- Administrador importa registro de transferencia.
- Administrador consulta reportes.
- Administrador genera ficha de pago.
- Administrador modifica la fecha de pago.
- Administrador modifica el pago.
- Administrador descarga historial de pago.
- Administrador registra la referencia personalizada.
- Administrador registra una solicitud de cobro.
- Administrador registra pago en efectivo.


Diagrama de Casos de Uso





Tabla de Priorización de Requisitos

Función	Riesgo	Valor	Complejidad	Estabilidad	Total
Inicia sesión	5	5	3	2	15
Consultar historial de pagos	5	4	3	1	13
Consultar estado de cuenta colegiatura	3	5	5	1	14

Consultar estado de cuenta otros servicios	3	3	5	1	12
Realizar pago en línea	5	5	2	4	16
Importar registro de transferencia	4	4	3	2	13
Consultar reportes	2	2	1	2	7
Generar ficha de pago	2	4	3	4	13
Modificar pago	3	4	3	3	13
Registrar referencia personalizada	5	3	2	2	12
Registrar solicitud de cobro	3	2	1	2	8
Registrar pago en efectivo	3	4	4	2	13
Descargar historial de pago	2	3	3	3	11
Modificar fecha de pago	3	4	3	3	13

 = *baja prioridad*

 = *media prioridad*

 = *alta prioridad*

Requisitos funcionales

Detalle de los casos de uso

CU01: Iniciar sesión

Descripción: Este caso de uso describe el proceso mediante el cual un usuario registrado inicia sesión en el sistema.

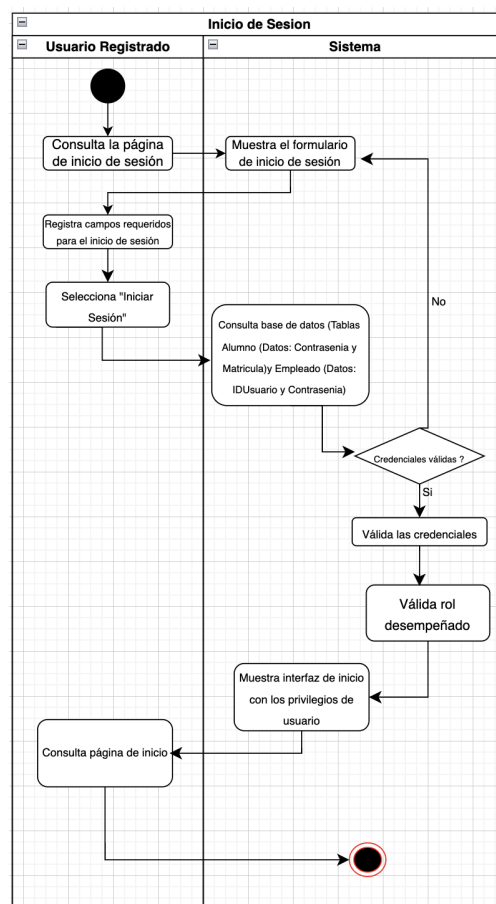
Actores: Usuario registrado

Precondiciones: El usuario debe estar registrado en el sistema.

Flujo Básico:

- El usuario accede a la página de inicio de sesión.
- El sistema muestra interfaz de inicio de sesión solicitando al usuario que ingrese su número de usuario y contraseña.
- El usuario ingresa su número de usuario y contraseña.
- El usuario selecciona Iniciar Sesión.
- El sistema consulta la base de datos (Tabla Alumno (Datos: Contraseña y Matrícula) y Empleado (IDUsuario y Contraseña))
- El sistema valida las credenciales del usuario.
 - Credenciales no válidas, el sistema muestra mensaje de error y redirige a la página principal de Inicio de Sesión.
- El sistema verifica el rol desempeñado por el usuario (Alumno, Administrador o Coordinador).
- El sistema muestra interfaz de inicio con los privilegios de usuario.
- El usuario consulta página de inicio.

Diagrama de Actividad



Postcondiciones:

- El usuario inicia sesión en el sistema y se otorgan los privilegios.

Caso de Uso: Consultar Historial de Pagos (Req 2.5)

Descripción: Este caso de uso describe el proceso mediante el cual un usuario registrado consulta su historial de pagos en el sistema.

Actores: Administrador.

Precondiciones:

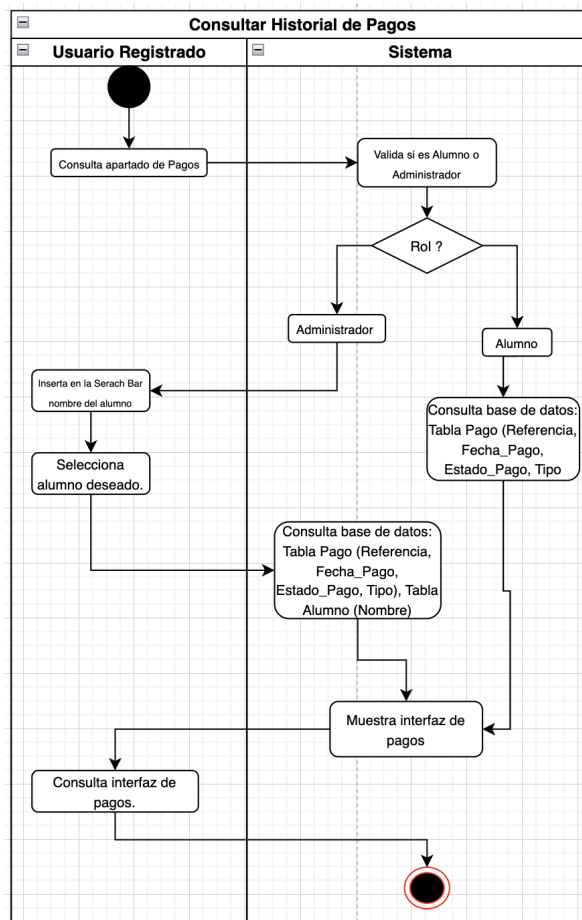
- El usuario debe estar registrado en el sistema.
- Debe existir un historial de pagos asociado al Alumno.

Flujo Básico:

- El administrador accede a la sección de pagos.
- El sistema muestra interfaz de Pagos.
- El sistema valida el rol que desempeña el usuario.
 - Si el usuario es alumno: Consultará la base de datos: Tabla Pagos (Referencia, Fecha_Pago, Estado_Pago, Tipo)
 - Si el usuario es administrador: Consultará de la base de datos: Tabla Alumno (Nombre), Tabla Pagos (Referencia, Fecha_Pago, Estado_Pago, Tipo)
- El sistema muestra interfaz con la información requerida.
- El usuario consulta la interfaz de Historial de Pagos.

Postcondiciones: El usuario visualiza su historial de pagos en el sistema.

Diagrama de Actividad



Caso de Uso: Realizar Pago en Línea (Req 3.2)

Descripción: Este caso de uso describe el proceso mediante el cual un alumno realiza un pago en línea a través del sistema bancario, incluyendo el registro de sus datos bancarios.

Actores:

- Alumno
- API de Santander

Precondiciones:

- El alumno debe estar registrado en el sistema.
- Debe existir un plan de pagos seleccionado por el alumno.

Flujo Básico:

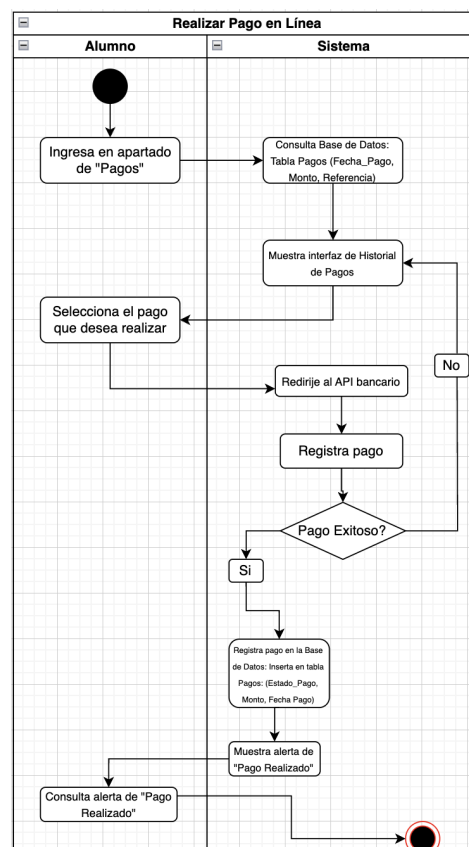
- El alumno ingresa en el apartado de “Pagos”.
- El sistema consulta base de datos: Tabla Pagos (Fecha_Pago, Monto, Referencia).

- El sistema muestra interfaz de Historial de Pagos.
- El alumno selecciona el pago que desea realizar.
- El sistema redirige a la API bancaria.
- El sistema registra el pago realizado.
 - Si el pago no fue realizado con éxito, el sistema redirigirá a la interfaz de Historial de pagos.
 - Si el pago se completó con éxito el sistema mostrará alerta de "Pago Realizado".
- El alumno consulta página de Historial de Pagos.

Postcondiciones:

- Se registra el pago en el sistema si la transacción es aprobada.
- El estado de la transacción se actualiza.

Diagrama de Actividad



Caso de Uso: Modificar fecha de pago (Req 4.4)

Descripción: En este caso de uso, el administrador podrá hacer uso de sus privilegios, para modificar la fecha en la que un pago puede ser hecho.

Actores:

- Administrador

Precondiciones:

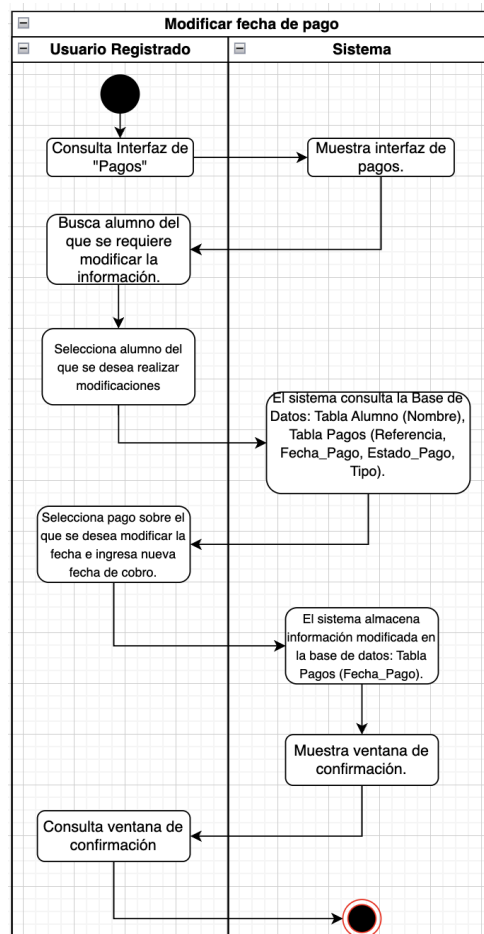
- El administrador debe de estar autenticado en el sistema y además debe de tener una sesión activa.

Flujo Básico:

- El administrador ingresa en el apartado de pagos.
- El sistema muestra interfaz de “Pagos”
- El administrador busca al alumno del que se desea consultar la información.
- El administrador selecciona el alumno del que desea hacer cambios.
- El sistema consulta la Base de Datos: Tabla Alumno (Nombre), Tabla Pagos (Referencia, Fecha_Pago, Estado_Pago, Tipo).
- El administrador selecciona el pago sobre el que desea modificar la fecha de pago e ingresa nueva fecha de cobro.
- El sistema almacena la nueva información en la base de datos: Tabla Pagos (Fecha_Pago).
- El sistema muestra ventana de confirmación.
- El administrador visualiza ventana de confirmación.

Postcondiciones:

- El sistema guarda la información en la base de datos y la muestra al alumno correspondiente.



Caso de Uso: Modificar pago (Req 4.4)

Descripción: En este caso de uso se describe el proceso por el cual el administrador modifica la cantidad a pagar de un alumno.

Actores:

- Administrador

Precondiciones:

- El administrador debe de estar autenticado en el sistema y además debe de tener una sesión activa.

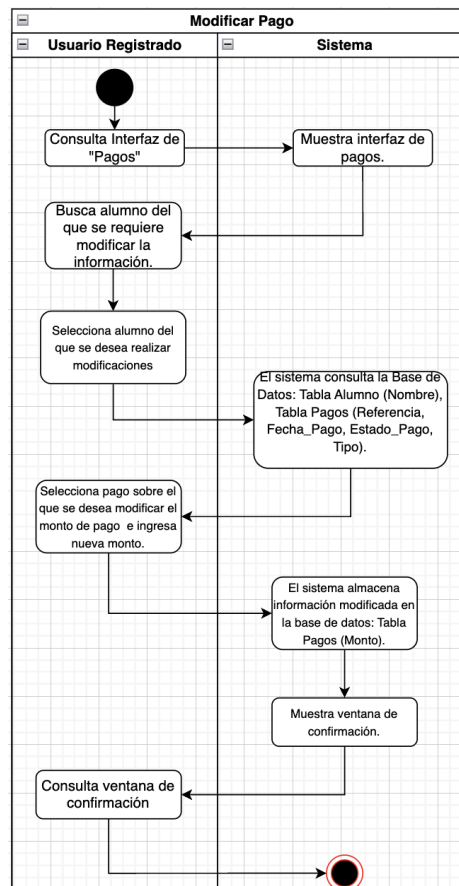
Flujo Básico:

- El administrador ingresa en el apartado de pagos.
- El sistema muestra interfaz de "Pagos"
- El administrador busca al alumno del que se desea consultar la información.
- El administrador selecciona el alumno del que desea hacer cambios.

- El sistema consulta la Base de Datos: Tabla Alumno (Nombre), Tabla Pagos (Referencia, Fecha_Pago, Estado_Pago, Tipo).
- El administrador selecciona el pago sobre el que desea modificar el monto e ingresa nuevo monto.
- El sistema almacena la nueva información en la base de datos: Tabla Pagos (Monto).
- El sistema muestra ventana de confirmación.
- El administrador consulta la ventana de confirmación.

Postcondiciones:

- El sistema guarda la información en la base de datos y la muestra al alumno correspondiente.



Caso de Uso: Registrar pago en efectivo (Req 4.8)

Descripción: En este caso de uso se describe el proceso por el cual el administrador puede registrar un pago hecho en efectivo.

Actores:

- Administrador

Precondiciones:

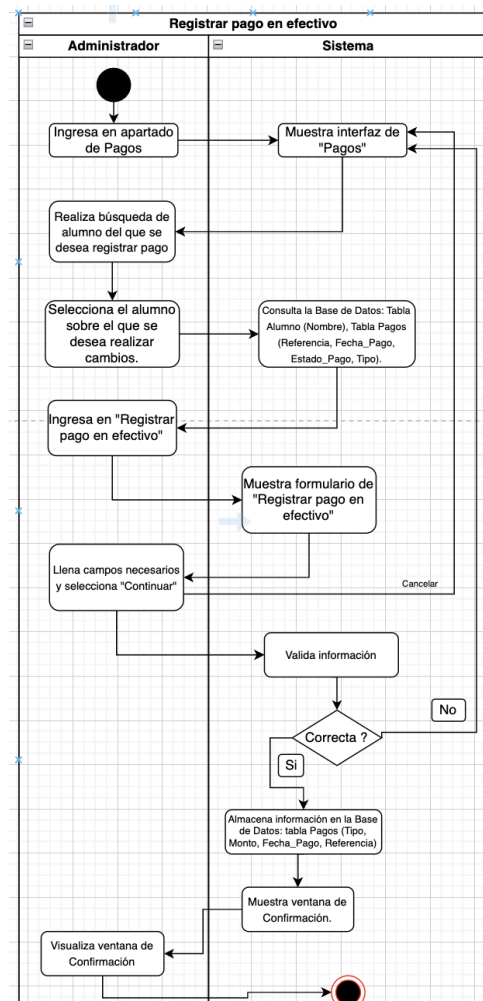
- El administrador debe de estar autenticado en el sistema y además debe de tener una sesión activa.

Flujo Básico:

- El administrador ingresa en el apartado de pagos.
- El sistema muestra interfaz de "Pagos"
- El administrador busca al alumno del que se desea consultar la información.
- El administrador selecciona el alumno del que desea hacer cambios.
- El sistema consulta la Base de Datos: Tabla Alumno (Nombre), Tabla Pagos (Referencia, Fecha_Pago, Estado_Pago, Tipo).
- El administrador ingresa en "Registrar pago en efectivo".
- El sistema muestra el formulario de "Registrar pago en efectivo" .
- El administrador rellena campos necesarios y selecciona "Continuar".
- El sistema valida la información proporcionada por el administrador.
 - Si la información es correcta, el sistema almacena información en la base de datos: Tabla Pagos (Tipo, Monto, Fecha_Pago, Referencia).
 - Si la información no es correcta, el sistema retorna a la interfaz de Pagos.
- El sistema muestra ventana de Confirmación.
- El administrador visualiza ventana de confirmación.

Postcondiciones:

- El sistema guarda la información en la base de datos y la muestra al alumno correspondiente.



Caso de Uso: Importar Registros de Transferencias (Req 3.3)

Descripción: Este caso de uso describe el proceso mediante el cual un administrador importa registros de transferencias bancarias realizadas por los alumnos.

Actores: Administrador

Precondiciones: El administrador debe tener los registros de transferencias disponibles en un formato compatible con el sistema.

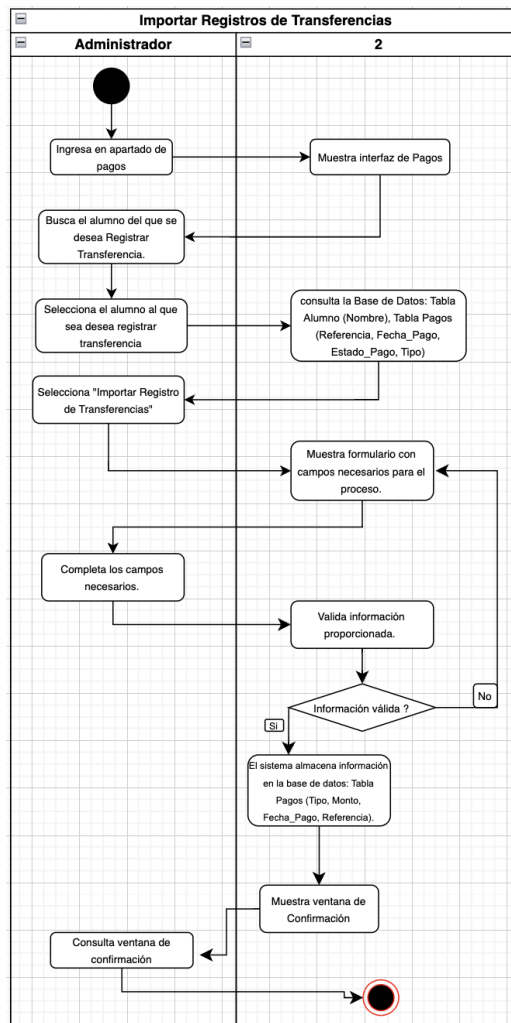
Flujo Básico:

- El administrador ingresa en el apartado de pagos.
- El sistema muestra interfaz de “Pagos”
- El administrador busca al alumno del que se desea consultar la información.
- El administrador selecciona el alumno del que desea hacer cambios.
- El sistema consulta la Base de Datos: Tabla Alumno (Nombre), Tabla Pagos (Referencia, Fecha_Pago, Estado_Pago, Tipo).
- El administrador ingresa en “Importar Registro de Transferencia”

- El sistema muestra Formulario con campos necesarios para Registrar la transferencia.
- El administrador completa los campos necesarios.
- El sistema valida la información proporcionada por el administrador.
 - Si la información es correcta, el sistema almacena información en la base de datos: Tabla Pagos (Tipo, Monto, Fecha_Pago, Referencia).
 - Si la información no es correcta, el sistema retorna a la interfaz de Pagos.
- El sistema muestra ventana de Confirmación.
- El administrador visualiza ventana de confirmación.

Postcondiciones: Los registros de transferencias se cargan en el sistema y están disponibles para su visualización y procesamiento.

Diagrama de Actividad



Caso de Uso: Generar Ficha de Pago

Descripción: Este caso de uso describe el proceso mediante el cual un administrador genera una ficha de pago para un alumno.

Actores: Administrador

Precondiciones:

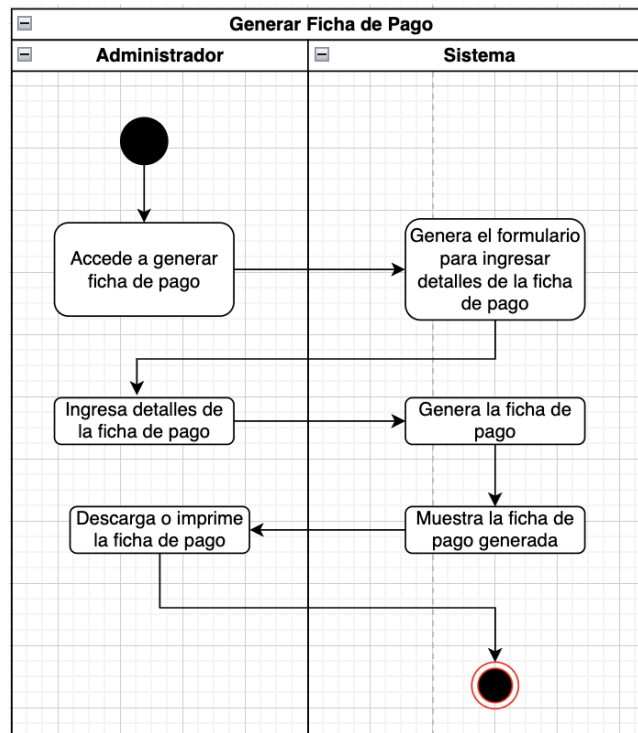
- El administrador debe tener acceso a la funcionalidad de generación de fichas de pago.
- Debe existir un alumno para el cual se desea generar la ficha de pago.

Flujo Básico:

- El administrador accede a la funcionalidad de generar ficha de pago.
- El sistema muestra un formulario para ingresar los detalles de la ficha de pago, como el nombre del alumno y el monto a pagar.
- El administrador ingresa los detalles de la ficha de pago.
- El sistema genera la ficha de pago con los detalles proporcionados.
- El sistema muestra la ficha de pago generada al administrador.
- El administrador guarda o imprime la ficha de pago según sea necesario.

Postcondiciones: Se genera una ficha de pago para el alumno especificado.

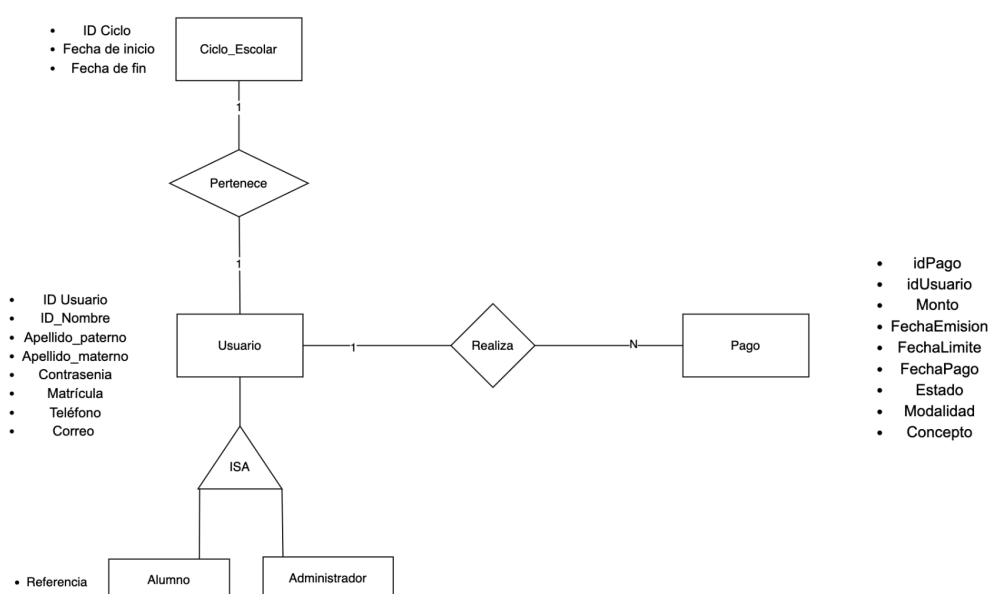
Diagrama de Actividad



Reglas de negocio

- El alumno tiene la opción de realizar el pago del semestre en una sola exhibición o bien dividirlo en 6 mensualidades.
- Si el alumno decide realizar el pago en 6 mensualidades, el primer pago será equivalente al 23% del pago total, mientras que el 77% restante se dividirá en 5 pagos equivalentes.
- Si el alumno cuenta con algún tipo de beca, el costo total del semestre se dividirá en 6 pagos iguales.
- En caso de que el alumno realice un pago mayor a lo correspondiente, este mismo será bonificado a la cuenta final.
- Si el alumno realiza el pago posteriormente al día indicado, se penalizará con un 5% adicional al monto correspondiente.
- El costo del crédito puede variar con el tiempo, debido a decisiones previamente consideradas por los directivos de la institución.
- Se contará con dos tipos de pagos “Colegiatura” y “Otros servicios”
- Los diplomados son considerados como “Otros servicios”
- Los alumnos deben de ser notificados acerca de un atraso de un mes, mientras que si el atraso es mayor o equivalente a dos meses, se marcará o resaltará a los alumnos con atraso en pagos.

Modelo Entidad-Relación (MER)



Diccionario de datos

DD_SOFT

Documentación de restricciones adicionales

Usuario:

- idUsuario: debe ser autoincrementable y es definido por la base de datos.
- Correo: es la matrícula del usuario más el dominio de la institución @ivd.com.
- Matrícula: será definida por el administrador.
- Referencia: será la matrícula más un número adicional, el cual lo asigna y podrá editar el administrador.

Pago:

- idPago: es autoincrementable y definido por la base de datos.
- idUsuario: debe ser el mismo identificador que se encuentra en Usuario.
- fechaEmision: será establecida por el administrador de acuerdo al día al que el pago sea establecido.
- fechaLimite: será definida por el administrador (puede ser modificada).
- fechaPago: será establecida de acuerdo a la realización de la fecha del último pago realizado.
- Estado: será definido de acuerdo al estado en que se encuentre el pago (aprobado, pendiente).
- Modalidad: será definida por el método en el que se realizó el pago (transferencia, en línea, efectivo).
- Concepto: concepto del pago, el cual será definido por el administrador.

Ciclo

- IDCiclo: será definido por el administrador.
- Fecha_inicio: será establecida por el administrador.
- Fecha_fin: será establecida por el administrador.

Tablas correspondientes (Modelo Relacional)

- **Entidad**
- **Llave Primaria**
- **Llave foránea**

Usuario(IDUsuario, Nombre, Apellido_paterno, Apellido_Materno, Correo, Contraseña, Matrícula, Teléfono, Referencia)

PK(IDUsuario)

FK (IDCiclo)

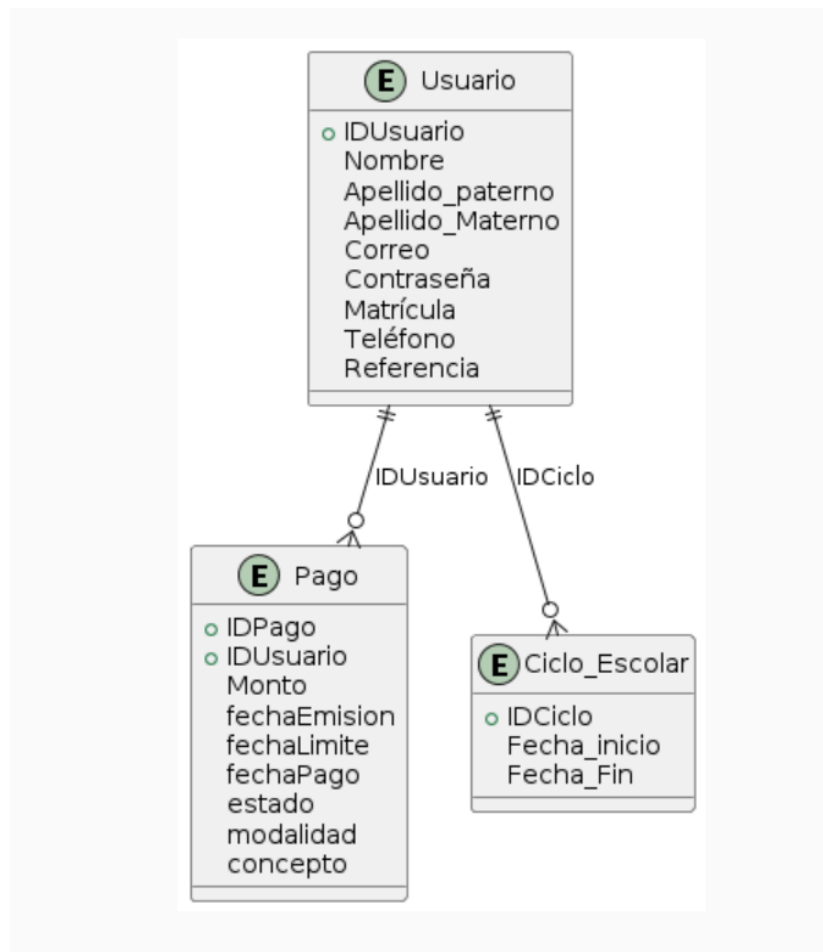
Pago(IDPago, IDPago, idUsuario, Monto, fechaEmision, fechaLimite, fechaPago, estado, modalidad, concepto)

PK(IDPago)

FK (IDUsuario)

Ciclo_Escolar(IDCiclo, Fecha_inicio, Fecha_Fin)

PK(IDCiclo)



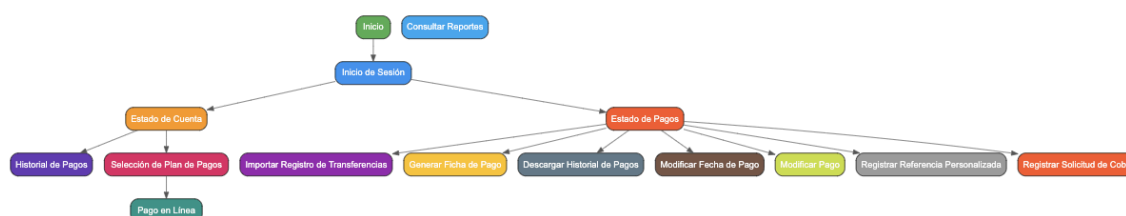
Requisitos no funcionales

- Usabilidad:
 - Diseñar una interfaz usuario intuitiva y visualmente atractiva, alineada con el estilo visual de Vía Diseño.
 - Conducir evaluaciones de usabilidad con usuarios finales para descubrir y corregir áreas de mejora en la interfaz.
 - Ofrecer una experiencia de usuario fluida a través de una navegación lógica y funciones debidamente identificadas.

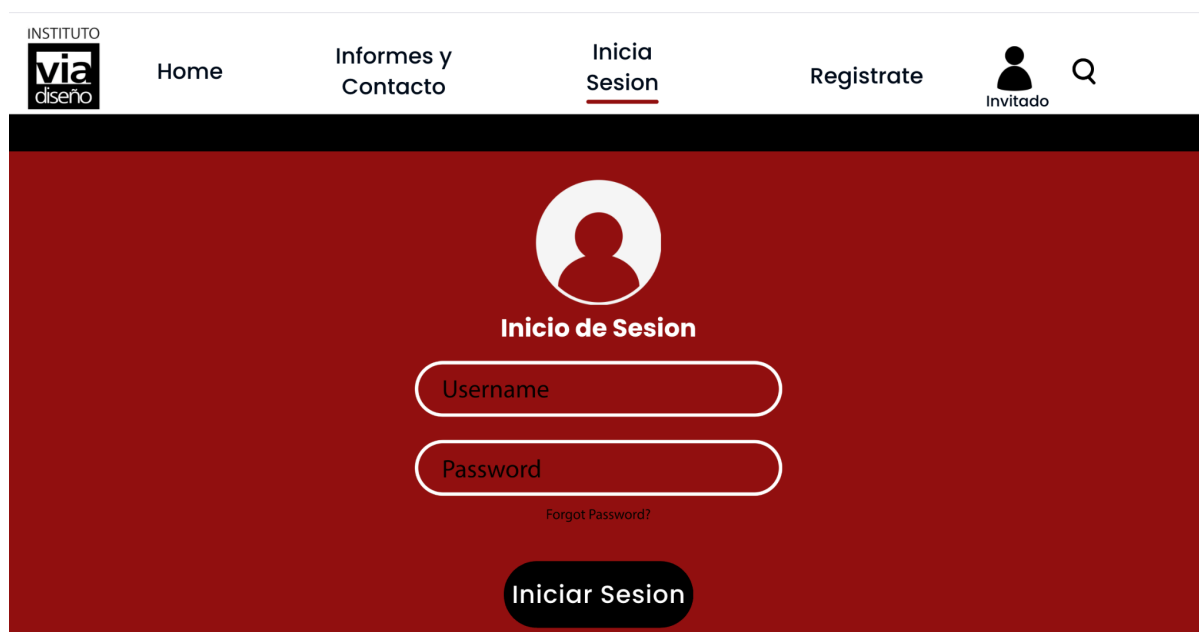
- Rendimiento:
 - Garantizar tiempos de respuesta rápidos y una alta capacidad de procesamiento de datos, especialmente para información sensible.
 - Optimizar código y estructura del sistema para acelerar los tiempos de respuesta.
 - Emplear tecnologías avanzadas para el procesamiento eficaz de grandes volúmenes de datos y utilizar técnicas de almacenamiento en caché.
 - Supervisar y ajustar continuamente el rendimiento para mantener niveles óptimos.
- Disponibilidad:
 - Asegurar una alta disponibilidad capaz de soportar la carga diaria de hasta 130 usuarios sin afectar el rendimiento.
 - Implementar balanceadores de carga y redundancia de componentes críticos.
 - Desarrollar un plan de continuidad operativa que incluya procedimientos de copia de seguridad y restauración.
 - Llevar a cabo mantenimiento preventivo regular.
- Seguridad:
 - Regular el acceso mediante un sistema de control basado en roles (administrador, coordinador, alumno).
 - Exigir políticas de contraseñas fuertes y registrar actividades de usuarios con privilegios.
 - Aplicar técnicas de cifrado avanzadas para proteger datos sensibles.
- Mantenibilidad:
 - Facilitar la comunicación efectiva y adaptación a cambios mediante un plan de comunicación y documentación actualizada del sistema.
 - Implementar un sistema de gestión de cambios y un proceso de revisión de código entre pares.
 - Realizar pruebas para asegurar la estabilidad del sistema tras actualizaciones.
- Escalabilidad:

- Permitir la escalabilidad vertical para adaptarse al crecimiento de la institución.
- Interoperabilidad:
 - Mantener una comunicación segura y eficiente entre la base de datos del portal administrativo, el portal de pagos y el API de Santander.
- Cumplimiento Normativo:
 - Cumplir con los requisitos legales de seguridad y confidencialidad de datos.
 - Establecer políticas y procedimientos para la protección de datos sensibles y mantener un registro de cumplimiento normativo.


Mapa del sitio



Bosquejo de la aplicación



Estado de Cuenta

Historial de Pagos 

Selección de Plan 

Plan semestral

Colegiatura	\$4,300.00
Otros Servicios	\$2,500.00
<hr/>	
Total:	\$23,500.00

Regresar

Continuar

Historial de Pagos

Fecha	Concepto	Monto
03/04/2024	Pago Colegiatura	\$5,000.00
05/10/2023	Pago Diplomado	\$2,000.00
01/02/2024	Pago Colegiatura	\$6,500.00

Regresar

Continuar

Pago en Línea

Selecciona metodo de pago

- **Tarjeta**
- **Transferencia Bancaria**
- **Paypal**

Monto a pagar: \$5,500.00

Regresar

Continuar

Administración y Reportes

Seleccione una accion

Consultar Reportes y Generar Ficha de Pago

Modificar Fecha de Pago

Registrar Referencia Personalizada

Modificar Pago

Descargar Historial de Pagos

Registrar Solicitud de Cobro

Regresar

Continuar

Plan de comunicación

Interesados en el proyecto:

- Softvolution
- Socios formadores
 - Adrian Gomez Romero
 - Bernardo Gomez Romero
- Usuarios finales del sistema
- Profesores

Interesado	Entregable	Frecuencia	Propietario	Prioridad	Manera de entrega
Softvolution	Avances semanales	Lunes, jueves, viernes (1:00 pm a 4:00 pm) Miércoles (9:00 am a 1:00 pm)	Ayetza Guadalupe Miguel	Muy prioritario	- Zoom - Sesiones presenciales.
Via Diseño	Estado de avance	Martes y jueves (3:00 pm a 4:00 pm) Viernes (9:00 am a 11:00 am)	Adrián Gomez Romero Bernardo Gómez Romero	Muy prioritario	- Slack - Mail - Sesiones presenciales - Zoom
Profesores	Estado de avance y consulta de dudas	Lunes, Martes, Jueves y Viernes (9:00 am a 1:20 pm)	Ricardo Cortés Espinoza Eduardo Daniel Juárez Pineda	Prioritario	- Slack - Mail - Sesiones presenciales - Zoom
Interesados Finales	Sistema finalizado	NA	NA	Prioritario	- Por medio del socio formador.

Vías de comunicación:

- Slack general (Profesores, alumnos de TC2005b)
- Slack de equipo (Profesores, Softvolution, Socio formadores)
- Grupo de Whatsapp con socios formadores.
- Correo electrónico, comunicación entre integrantes de equipo, socio formadores y profesores de la clase TC2005B.
 - Integrantes de Equipo
 - Ayetza Yunnuen Infante García: A01709011@tec.mx
 - Maria Guadalupe Soria Velazquez: A01710797@tec.mx
 - Miguel Angel Becerra Ayala: A01710076@tec.mx

- Slack del equipo:
<https://app.slack.com/client/T08BWSY0P/C06TG7VAQ0Z>
- Socios formadores
 - Adrián Gómez Romero: adriangomezr95@gmail.com
 - Bernardo Gómez Romero: bernardogr95@gmail.com
- Profesores
 - Ricardo Cortés Espinosa: rcortese@tec.mx
 - Eduardo Daniel Juárez Pineda: edjuarezp@tec.mx

Guía de estilo de codificación

En nuestro proyecto, adoptaremos una guía de estilo basada en las convenciones de la comunidad y en parte en las recomendaciones de la Guía de Estilo de JavaScript de Airbnb, que es una guía de estilo de terceros ampliamente aceptada en la comunidad de desarrollo de JavaScript.

1) Variables:

- a) Utilizaremos nombres descriptivos y significativos para las variables.
- b) Seguiremos la convención camelCase para nombres de variables.
- c) Evitaremos nombres de variables demasiado cortos o ambiguos.
- d) Utilizaremos `const` para valores que no cambiarán y `let` para valores que puedan cambiar.

2) Strings:

- a) Usaremos comillas simples (`'`) para strings en general, y comillas dobles (`"`) para strings que necesiten interpolación de variables.

3) Funciones:

- a) Utilizaremos nombres descriptivos y claros para las funciones.
- b) Utilizaremos la convención camelCase para nombres de funciones.
- c) Limitaremos la longitud de una función, preferiblemente menos de 20 líneas.
- d) Evitaremos funciones que realicen múltiples tareas, siguiendo el principio de responsabilidad única.

4) Parámetros de Funciones:

- a) Utilizaremos nombres descriptivos para los parámetros de funciones.
- b) Evitaremos nombres de parámetros demasiado cortos o ambiguos.

- c) Evitaremos la sobrecarga de parámetros; preferiblemente, pasaremos un objeto si hay demasiados parámetros.
- 5) Comentarios:
- a) Utilizaremos comentarios para explicar el código cuando sea necesario, especialmente en partes que puedan ser difíciles de entender.
 - b) Documentaremos funciones y métodos usando el formato de JSDoc.
- 6) Objetos:
- a) Utilizaremos nombres descriptivos y significativos para las propiedades de los objetos.
 - b) Seguiremos la convención camelCase para los nombres de las propiedades de los objetos.
 - c) Utilizaremos getters y setters solo cuando sea necesario; preferiblemente, utilizaremos acceso directo a las propiedades.
- 7) Estatutos de Control:
- a) Utilizaremos una indentación consistente, preferiblemente de 2 o 4 espacios.
 - b) Usaremos llaves `{}` incluso para bloques de código de una sola línea en estatutos de control (if, for, while) para mayor claridad.
 - c) Dejaremos un espacio después de palabras clave como `if`, `else`, `for`, `while`, etc.

Plan de trabajo actualizado

Plan de Trabajo

Aprendizaje adquirido

Elabora un video de máximo 5 min sobre el avance de tu proyecto. Asegura que los profesores y tu socio formador tengan acceso al video. Pon el enlace al video en el Readme.md del repositorio de tu proyecto.

Antes de realizar la entrega o presentación del avance, en equipo deberán realizar y entregar firmada su coevaluación a los profesores.

Incumplir este aspecto, equivale a 0 puntos para todos los miembros del equipo en esta entrega.