

FACULTAD:	Tecnología Informática		
CARRERA:	Analista Programador		
ALUMNO/A:	ROLANDO BETIANA		
SEDE:	Buenos Aires	LOCALIZACIÓN:	Distancia
ASIGNATURA:	Metodología de Desarrollo de Sistemas II.		
CURSO:	T4-17-15 2-K-D	TURNO:	Distancia
PROFESOR:	Leonel Jiménez G.	FECHA:	27/05/2023
TIEMPO DE RESOLUCIÓN:	5 días.	EXAMEN PARCIAL NRO:	2
MODALIDAD DE RESOLUCIÓN:	Domiciliario – Individual.		
Resultados de aprendizaje:			
<ul style="list-style-type: none"><li>[Diseña]+ [El sistema informático] + [Para representar la estructura estática y el comportamiento dinámico] + [Aplicando un Proceso de desarrollo] / [integrando todas las vistas de modelado] / [Aplicando el estándar UML]</li><li>[Resuelve]+ [en posibles situaciones de conflicto]+ [para consensuar ideas comunes] +[considerando las características de cada participando del grupo]</li><li>[Elabora] + [Informes técnicos]+ [para comunicar sus producciones] + [considerando la legibilidad y organización de la información]</li></ul>			

Los estudiantes encontrará el examen habilitado en la plataforma ULTRA a partir de la fecha y hora indicada por el profesor.

**Criterios de calificación:** Para acreditar los saberes deberá tener aprobado, al menos, el 60% de los aspectos conceptuales, además de tener aprobado también, al menos, el 60% de los aspectos procedimentales. La calificación final se obtendrá luego de la defensa oral del trabajo presentado.

**Criterios de evaluación:** Se evaluará la claridad en el planteamiento de los aspectos conceptuales y procedimentales. La evaluación se hará a partir de la siguiente grilla:

Criterio	Calificación	Observaciones
Instancia oral		
<b>Aspectos Conceptuales</b>		
Pregunta 1		
Pregunta 2		
Pregunta 3		
Pregunta 4		
Pregunta 5		
<b>Aspectos procedimentales</b>		
Pregunta 1		
Pregunta 2		
Pregunta 3		
Pregunta 4		
Calificación final		

### Forma de entrega del examen

- Se debera resolver el examen sobre el mismo archivo del examen original.
- Se deberá subir al ULTRA el examen ya resuelto en un archivo tipo ZIP o RAR.

### Aspectos conceptuales

1. Enumere y describa los pasos implicados dentro de la técnica (Artículo) “De los Procesos de Negocio a los Casos de Uso” [30]

***Identificación y modelado de procesos de negocio:*** Utilizamos técnicas como la observación y la colaboración con las partes interesadas para identificar y comprender los procesos de negocio de la organización. Analizamos detalladamente las actividades, roles, flujos de trabajo y responsabilidades involucradas. Definimos objetivos principales y los descomponemos en subobjetivos más específicos. Cada subobjetivo se convierte en un proceso de negocio que respalda su logro. Representamos estos procesos como casos de uso del negocio, descritos de manera clara. Luego, los modelamos utilizando diagramas de negocios y actividades para visualizar su flujo y estructura de forma comprensible. A través de este enfoque, valoramos el impacto humano de estos procesos en el éxito organizacional

***Identificación de actores y objetivos.*** Una vez que hemos identificado los procesos de negocio, nos adentramos en la búsqueda de las personas y entidades involucradas en su ejecución. Nos dedicamos a reconocer los roles

desempeñados tanto por miembros internos como externos de la organización. Luego, creamos un diagrama de casos de uso del negocio que nos permite visualizar y definir el entorno de la organización, destacando especialmente a los actores externos al sistema.

A continuación, nos sumergimos en la creación de diagramas que representan los diferentes roles y la secuencia de su participación. Durante este proceso, nos ponemos en los zapatos de los actores y buscamos comprender sus necesidades y objetivos. Esto nos ayuda a desarrollar representaciones claras y humanas de cómo interactúan y colaboran los diferentes actores en el contexto del negocio.

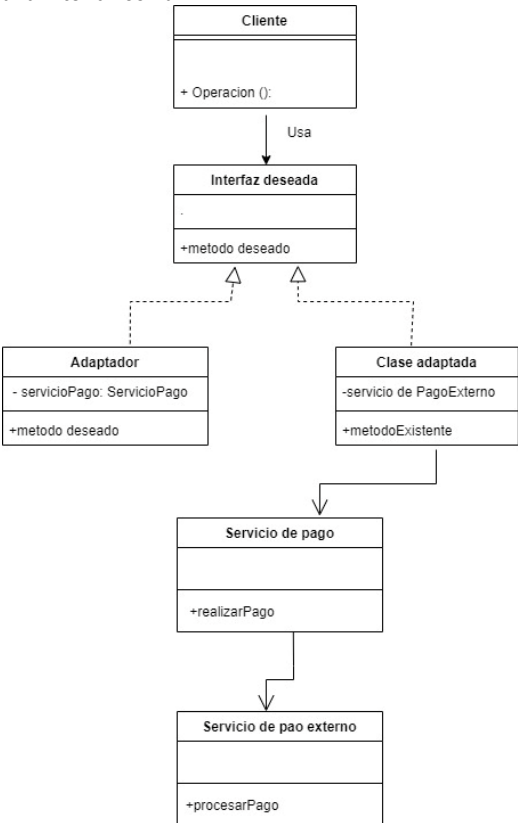
**Descripción de los Casos de Uso del Negocio:** En esta etapa, nos sumergimos en la descripción detallada de cada caso de uso del negocio. Analizamos cuidadosamente cada caso de uso y observamos los roles involucrados, tanto dentro como fuera de la organización. Nos enfocamos en comprender quiénes son los actores internos que desempeñan un papel en cada caso de uso. A medida que nos sumergimos en la descripción de cada caso de uso, nos ponemos en los zapatos de los usuarios y agentes involucrados. Buscamos comprender sus necesidades, desafíos y cómo interactúan con el sistema. Este enfoque humano nos ayuda a capturar de manera precisa y comprensible el flujo de trabajo y las interacciones en cada escenario de negocio

**Diagrama de Roles:** Creamos un diagrama que ilustra cómo los roles colaboran entre sí para llevar a cabo un caso de uso del negocio. Cada rol se representa como una clase UML especial, y mostramos las asociaciones entre los roles con los que pueden colaborar. Este diagrama nos brinda una visión clara y visualmente atractiva de la estructura de colaboración en el contexto del negocio

**Creación de Escenarios:** Nos adentramos en la creación de escenarios para representar el aspecto del comportamiento en la colaboración entre los roles. Utilizamos diagramas de secuencia UML, donde los objetos representan las instancias de los roles que participan en la interacción. En estos escenarios, nos sumergimos en situaciones concretas, narrando visualmente cómo se desarrolla la interacción entre los diferentes roles. Nos enfocamos en capturar los pasos, acciones y comunicación que ocurren entre ellos, brindando una representación comprensible y detallada de cómo se lleva a cabo la colaboración. A través de estos diagramas de secuencia, no solo visualizamos el flujo de acciones, sino que también damos vida a los roles, mostrando cómo interactúan, se comunican y trabajan juntos para alcanzar los objetivos del negocio. Este enfoque humano nos permite comprender de manera más profunda y significativa la dinámica de la colaboración entre los roles

2. Explique para qué se utiliza el patrón Adaptador. (Explicación y Diagrama de Clases) [20].

El patrón Adaptador se utiliza cuando tenemos dos clases o componentes que no se pueden comunicar directamente debido a diferencias en sus interfaces. Un cliente que necesita interactuar con una clase existente, pero sus interfaces no son compatibles. En ese caso, el adaptador actúa como un intermediario o "traductor" entre el cliente y la clase existente. El adaptador implementa la interfaz que el cliente espera utilizar y se encarga de adaptar o convertir las llamadas del cliente en llamadas adecuadas para la clase existente. El diagrama de clases muestra al cliente, la interfaz deseada por el cliente, el adaptador y la clase existente. El adaptador se conecta entre el cliente y la clase existente, permitiendo que ambos se comuniquen a través de una interfaz común.



3. Explique las siguientes expresiones en OCL:

- Context Proyecto inv: self.participa -> select (edad > 50) ->notEmpty() [5].

- Esta expresión se utiliza para definir una restricción en el contexto de la clase "Proyecto"., esta expresión indica una restricción que indica que debe haber al menos una entidad(objeto) relacionada con el proyecto cuya edad sea mayor a 50 años.
- Context Job inv: self.empleado.edad > 21 [5].

Esta expresión se utiliza para definir una restricción en el contexto de la clase "Job" (trabajo). La restricción indica que en el contexto de un trabajo, la edad del empleado asociado a ese trabajo, representada por la propiedad "empleado.edad", debe ser mayor a 21 años.

Estas expresiones en OCL se utilizan para definir restricciones o condiciones que deben cumplirse en un modelo de datos. En ambos casos, se establecen condiciones relacionadas con la edad de las entidades involucradas.

4. Explique cómo se realiza paso a paso la transformación de un Diagrama de Clases de un Diagrama Entidad Relación. [20 Puntos]
  1. Identificar las clases y sus atributos en el Diagrama de Clases.
  2. Convertir cada clase en una entidad en el Diagrama Entidad-Relación (DER) y asignar los atributos correspondientes.
  3. Determinar las relaciones y sus cardinalidades entre las entidades en el DER.
  4. Establecer las claves primarias para cada entidad en el DER.
  5. Transformar propiedades multivaluadas en entidades separadas con relaciones de uno a muchos.
  6. Agregar restricciones de integridad referencial para mantener la consistencia de los datos.
  7. Revisar y ajustar el DER para asegurarse de que refleje correctamente la estructura y relaciones del sistema.
5. Explique el siguiente diagrama que se relaciona con las metodologías Ágiles [20].

El diagrama de Scrum representa un marco de trabajo utilizado en el desarrollo de software bajo la metodología ágil.

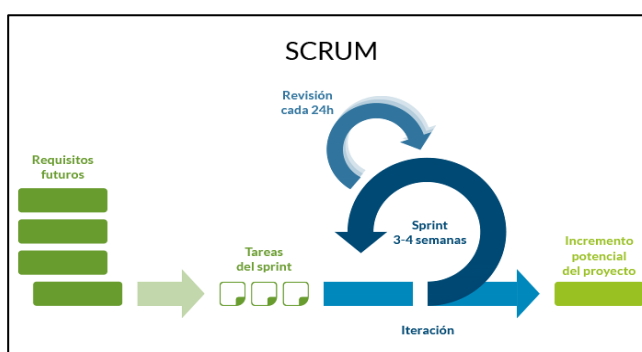
Tiene tres roles principales:

**Product Owner:** Representa los intereses y necesidades de los usuarios o clientes. Se encarga de gestionar una lista priorizada de requisitos y funcionalidades llamada Product Backlog.

**Scrum Master:** Facilita el trabajo del equipo y asegura que siga los principios y prácticas de Scrum. Elimina obstáculos y promueve la colaboración y mejora continua.

**Equipo de Desarrollo:** Desarrolla y entrega el producto. Toma decisiones y se organiza para alcanzar los objetivos del Sprint.

El diagrama Scrum muestra la estructura y el flujo de trabajo en un proceso Scrum. Comienza con el Product Backlog(requisitos), se avanza a través del Sprint Backlog(lista de tareas) y el Sprint de 3 a 4 semanas, se realiza una revisión del Sprint al final de cada iteración y se incorpora el feedback en futuros Sprints. La iteración continua de este proceso permite el desarrollo incremental del producto y la adaptación a medida que se avanza



## Aspectos procedimentales

### Para el siguiente Diagrama de Caso:

Se necesita construir el sistema de gestión del programa de viajeros frecuentes de una línea aérea. El programa de viajero frecuente de la línea aérea permite a sus clientes acumular millas que luego pueden cambiar por premios. Para esto, a cada cliente se le otorga una tarjeta. La tarjeta puede ser de Nivel Plata o Nivel Oro (inicialmente es de nivel plata). El sistema permite cargar los viajes que hacen los usuarios y actualizar el nivel de las tarjetas de los usuarios.

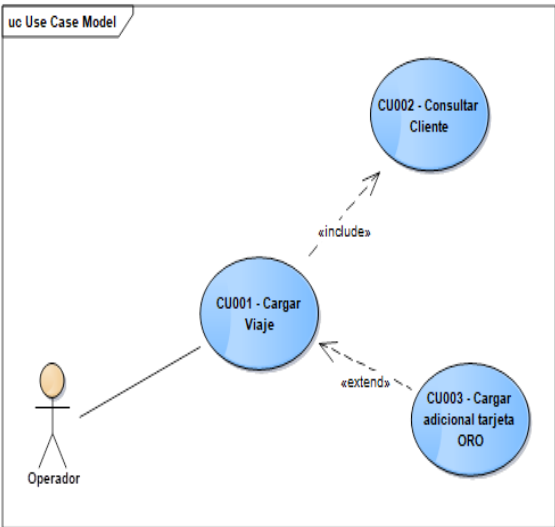
**Registrar usuario.** Cuando una persona se subscribe al programa, se registra de la misma: su nombre, nacionalidad, dirección postal y de email, y su número de pasaporte.

**Carga de viajes.** Cuando una persona hace un viaje, se registra del mismo; el origen, el destino, el número de vuelo, la fecha, y las millas voladas. Además, cada viaje suma millas de acuerdo al siguiente criterio:

- **Calculo base:** Simplemente se suman las millas del viaje. Si la persona tiene tarjeta oro se duplica la cantidad de millas del viaje.
- **Adicionales:** Los viajes de distancia de más de 10.000 millas suman 1.000 millas adicionales para la tarjeta plata y 2.000 para la tarjeta oro. Para la tarjeta oro, se suman 2.000 millas adicionales para todo viaje dentro del país.

**Cobro de premios.** El sistema permite registrar el cambio de puntos por distintos premios (viajes, electrodomésticos, etc). Por cada cambio registraremos la fecha, el premio elegido, la cantidad de puntos utilizados.

**Emisión de resúmenes.** El sistema permite emitir el resumen de la cuenta de cada usuario. Del resumen puede obtenerse, la fecha de emisión del mismo, el total de millas acumuladas desde el último balance que no han sido cambiadas por premios, el total de millas cambiadas por premio desde el ultimo balance, el total de millas que vencerán en el próximo balance (es decir, a un año del último balance).



Se Requiere:

1 Diagrama de Secuencia del CU 001 [20].

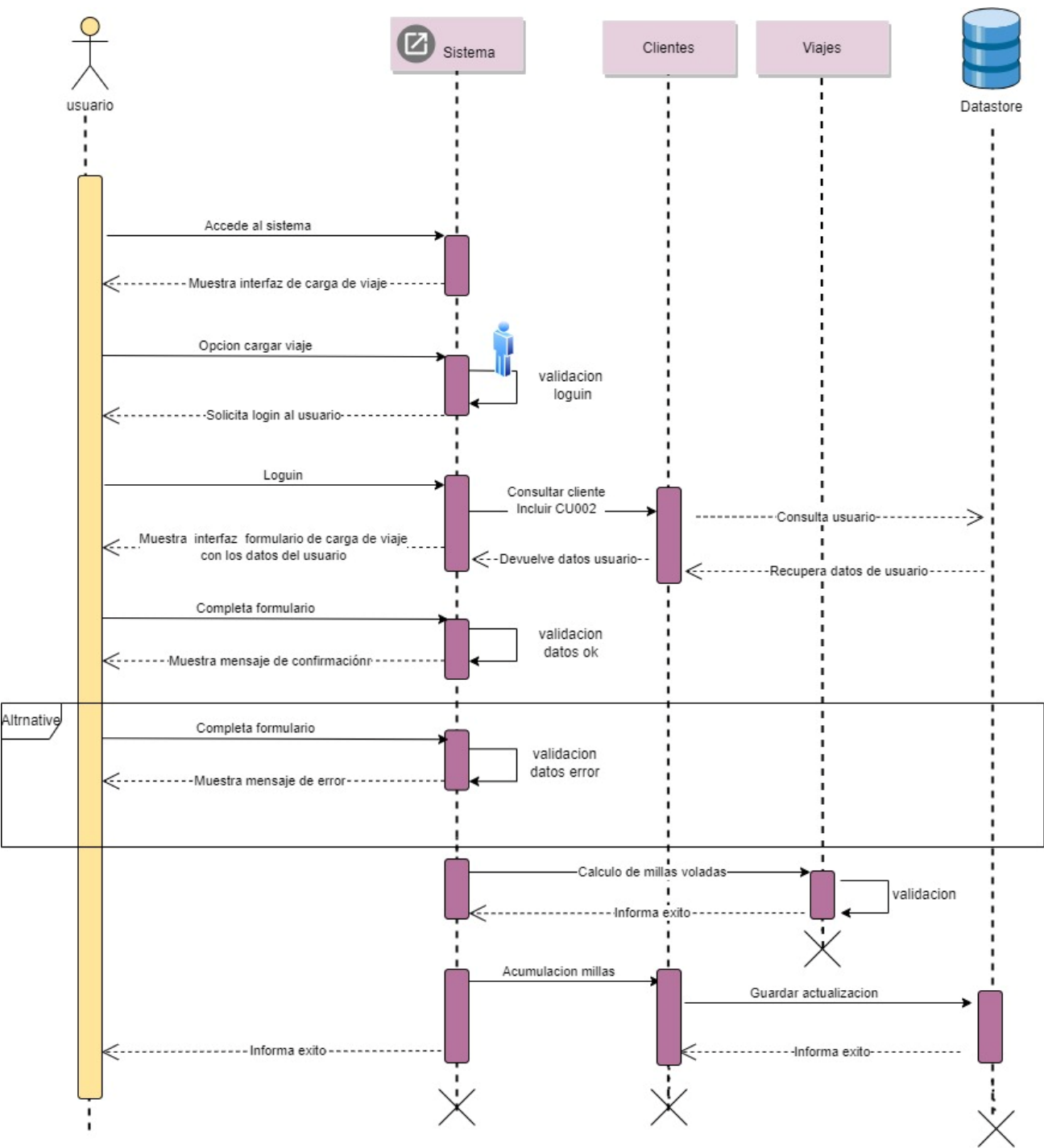
2 Diagrama de Actividad del CU 001 [20].

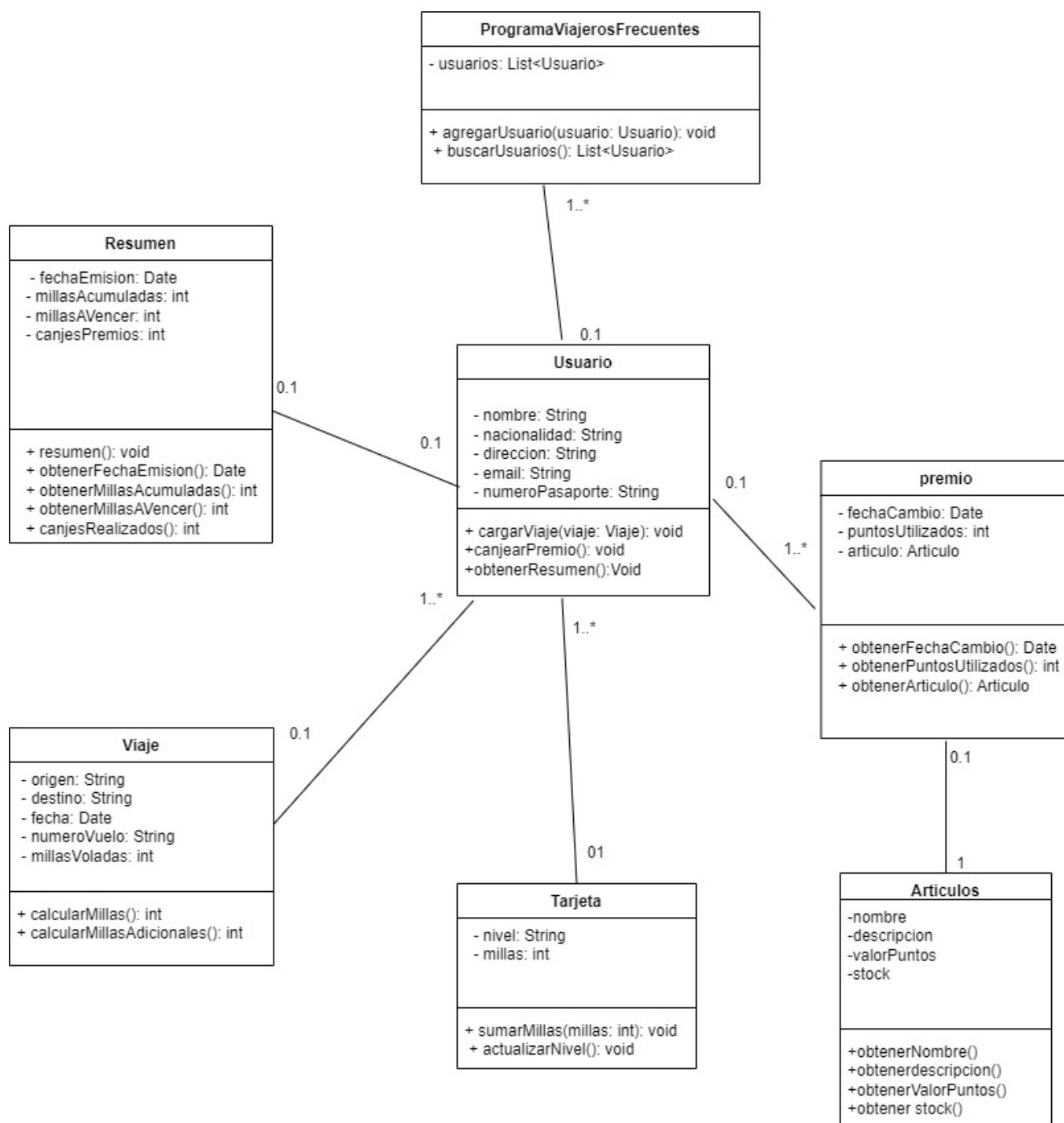
3 Realizar el Diagramas de clase de toda la aplicación e implementar dos restricciones por medio de OCL [30].

4 Realizar la conversión al modelo de datos correspondiente al punto 3 [30].

Se pueden adicionar más atributos de datos si se consideran convenientes, siempre y cuando se mantengan dentro de la lógica del tema y del sentido común. Cualquier asumisión adicional que se emplee debe ser descrita en el examen.

CU-001	CARGAR VIAJE	
Dependencias	Ninguna	
Precondición	El usuario pertenece al programa de millas.	
Descripción	Este caso de uso permite a un usuario cargar un nuevo viaje en el sistema, registrando la información relevante del viaje y actualizando las millas acumuladas en su cuenta.	
Secuencia Normal	Paso	Acción
	1	El Usuario accede al sistema.
	2	El sistema muestra interfaz de carga de viaje
	3	El Usuario selecciona la opción "Cargar Viaje" en la interfaz de usuario.
	4	El Sistema verifica que el Usuario esté logueado.
	5	El Sistema solicita el login al usuario
	6	El Usuario se loguea.
	5	El Sistema incluye el caso de uso "CU002: Consultar Cliente" para mostrar la información del cliente
	6	El Sistema interfaz formulario de carga de viaje con los datos del usuario.
	7	El Usuario completa los campos del formulario de carga de viaje (origen, destino, número de vuelo, fecha, millas voladas) y envía la información al Sistema.
	8	El Sistema valida los datos ingresados, verificando que los campos obligatorios estén completos y en el formato correcto. Envía mensaje de confirmación.
	9	El Sistema realiza el cálculo de millas acumuladas para el viaje, considerando el nivel de tarjeta del Usuario.
	10	El Sistema actualiza las millas acumuladas en la cuenta del Usuario.
11	El Sistema muestra un mensaje de confirmación al Usuario, informando que el viaje ha sido cargado exitosamente y mostrando las millas acumuladas actualizadas.	
Postcondición	El viaje se registra correctamente en el sistema. Las millas acumuladas en la cuenta del Usuario se actualizan.	
Excepciones	6.1	Si los datos ingresados en el formulario de carga de viaje son inválidos o incompletos, se muestra un mensaje de error. Fin del caso de uso
	10.1	Si el Usuario decide cambiar a Tarjeta Dorada, se sigue el flujo del caso de uso "CU003: Asignación Tarjeta Oro".





context Viaje  
inv: millasVoladas >= 0

context Usuario  
inv:  
self.nombre.size() > 0 and self.direccion.size() > 0 and self.email.size() > 0 and self.numeroPasaporte.size() > 0  
implies self.nacionalidad.size() > 0



```
ProgramaViajerosFrecuente(cod_programaViajerosFrecuentes, cod_usuario(Usuario))

Usuario (cod_usuario, cod_programaViajerosFrecuentes (ProgramaViajerosFrecuente))

Resumen (cod_resumen, cod_usuario, (Usuario), cod_programaViajerosFrecuentes(ProgramaViajerosFrecuente))

Viaje (cod_viaje, cod_usuario(usuario))

Premio (cod_premio, cod_usuario, (Usuario), cod_articulo, (articulo))

Articulo (cod_articulo)

Tarjeta (cod_tarjeta, cod_usuario, (Usuario))
```

