Obligatorio 2021

Tercera Entrega

Bases de datos

Integrantes del grupo:

- **Diego Kleinman**
- **Ignacio Calderazzo**
- **Federico Valiño**

Docentes:

Esteban Barrios

Bernardo Rytchenberg

Índice

1. Introducción

1.1 Propósito

El objetivo del presente documento es definir los requerimientos funcionales y no funcionales para el desarrollo de un sistema de trueques, llamado UCUTrade. El sistema será utilizado por alumnos y profesores.

1.2 Alcance

El documento está dirigido principalmente a los interesados y usuarios finales, en este caso los profesores del curso, de ahora en más, ***"stakeholders".*** También será de utilidad para los desarrolladores ya que brinda un entendimiento completo del sistema a desarrollar.

1.3 Ámbito del sistema: sistemas de tipo LETS

Los sistemas de tipo LETS (Local Exchange Trading Systems) funcionan como una red centralizada de intercambios en dónde es posible comercializar productos y servicios utilizando una moneda propia que tiene un valor dentro de la misma plataforma. El sistema no permite transacciones con monedas corrientes externas, por ejemplo: Euro, Peso Uruguayo, Dólar, etc.

LETS se consolida como un sistema de crédito de tipo mutualista, en los cuales no existe ánimo de lucro con respecto a las transacciones entre los participantes, ya que no existen intereses y recargos, porque se parte del principio del comercio voluntario, en dónde existirá confianza y reciprocidad voluntaria de entre partes. Debido a las características anteriormente mencionadas, en estos sistemas no existen fenómenos económicos como la inflación, deflación o tasas de interés.

Para que un sistema se pueda considerar de tipo LETS, debe que cumplir con los siguientes principios:

- Intereses: Los créditos de los asociados no generan interés.
- Consentimiento: Los asociados no están obligados a comerciar.
- Transparencia: Los asociados pueden ver los saldos de otros miembros de la plataforma.
- Equivalencia: Se toman referencias de otras monedas corrientes.
- Coste: Se suele cobrar alguna cantidad en dinero corriente para mantenimiento de la plataforma.

La mecánica de estos sistemas es muy sencilla, imaginemos que tenemos tres individuos que serán los asociados, a efectos de esta explicación se llamarán *Sujeto 1, Sujeto 2 y Sujeto 3*, todos se inscriben en un sistema de tipo LETS, partiendo todos con un saldo inicial, cambio y saldo final en 0.

El Sujeto 2 desea hacer una venta, por lo que realiza una publicación por un kilo de arroz a valor de 5 coins del sistema, el Sujeto 1 es quién termina realizando la compra. El balance quedará algo así:

	Sujeto 1	Sujeto 2	Sujeto 3
Saldo Inicial	0	0	0
Cambio	-5	5	0
Saldo Final	-5	5	0

Expliquemos a continuación la situación de ambos sujetos:

- Sujeto 2:
    - Posee un saldo positivo de cinco coins del sistema.
    - Los podrá utilizar para intercambiar artículos o servicios con cualquier persona asociada a este circuito (Es decir, Sujeto 1 y Sujeto 3).

- Sujeto 1:
    - Tendrá saldo negativo de cinco coins del sistema.
    - No tiene que pagar intereses o moras, debido a que esta es una moneda que no tiene fines de lucro.
    - Su obligación será ofrecer a cualquier otro socio el equivalente de lo que ha recibido, por ejemplo: un kilo de manzanas.

Algunos comentarios para tener en cuenta acerca de las reglas de un sistema LETS:

- Se suele fijar una cantidad máxima de saldo negativo, para así evitar que las personas compren y que no ofrezcan nada a cambio.
- La garantía es el compromiso y transparencia de los asociados para ofrecer algo de acuerdo al saldo positivo que este individuo tenga.
- La suma de los saldos de todas las cuentas siempre debe dar 0.
- Todos los asociados en este sistema LETS que hemos presentado, podrán ver una tabla con todos los saldos de todos los miembros.

Para resolver la problemática planteada en el obligatorio, se tomarán varias características de los sistemas LETS, con la principal diferencia de que el crédito únicamente se utilizará como valor de referencia para los artículos. A continuación se explicarán las características y restricciones que se incluirán en el sistema.

2. Descripción del sistema

2.1 Introducción al sistema

Este es un sistema de software independiente y desarrollado desde 0, el cuál estará orientado al ***"trading"*** de productos online. El producto que será desarrollado, contará únicamente con las funcionalidades básicas de este tipo de sistemas ya que se trata de un prototipo, es decir, una versión "***demo"*** del sistema con el fin de poder ser extendida a futuro.

Al tratarse de un sistema que involucra transacciones y estando basado en un sistema de tipo LETS, es necesario que el sistema cuente con un crédito interno, en este caso son las denominadas UCUCoins.

Estas coins a priori, tendrán un valor equivalente al precio de una botella de un litro de agua (aproximadamente unos $50 pesos uruguayos). Es importante mencionar que este valor monetario será de utilidad **únicamente** para que los usuarios puedan asignar el valor a los artículos que publican.

2.2 Funcionalidades del producto

2.2.1 Registro y Login de usuarios

Para utilizar el sistema, los usuarios deberán registrarse brindando sus datos: cédula, nombre, apellido, teléfono y correo. Además deberá elegir una contraseña para completar los datos de usuario.

Una vez configurado el nuevo usuario, tendrá acceso al sistema mediante un login.

2.2.2 Posteo de productos que el usuario quiere intercambiar

Los usuarios registrados podrán ***"postear"*** los productos (de carácter legal), que estén interesados en intercambiar por otros productos. Para realizar la publicación el usuario deberá adjuntar fotografías del producto, una descripción y elegir un valor en UCUCoins. Una vez confirmada la publicación, ésta quedará visible en el perfil del usuario para otros usuarios registrados.

2.2.3 Ofertar trueques

Los usuarios registrados podrán ver todas las publicaciones de los demás en la pantalla principal. Si el usuario está interesado en obtener uno de los productos vistos en el ***"feed"***, podrá realizar una oferta al usuario propietario de la publicación.

La oferta que se realice debe contener uno o varios de los productos publicados por el usuario que realiza la oferta y deben igualar el valor del producto que se desea obtener.

2.2.4 Confirmación o contraofertas de trueques

Los usuarios que reciban ofertas verán en su perfil, el detalle de las ofertas recibidas. Cuando el usuario ingrese a alguna de las ofertas recibidas, podrá ver qué producto (o productos) se le están ofreciendo a cambio de que productos de los que él mismo oferta.

Dentro de una oferta recibida, el usuario podrá aceptar, rechazar o realizar una contraoferta. Si la oferta es aceptada, los usuarios involucrados en la transacción podrán ver los datos de contacto del otro con el fin de concretar el trueque. En caso de que la oferta sea rechazada, se le notificará al usuario que realizó la oferta y además se eliminará del perfil de quién recibió la oferta.

Por otra parte, se podrán realizar contraofertas. Si el usuario elige esta opción, podrá ver el perfil del usuario que realizó la oferta con el fin de ver sus otras publicaciones existentes. Aquí, el usuario que ejecuta la contraoferta podrá elegir cualquiera de los productos publicados por el usuario que originalmente realizó la oferta, para incluir en la transacción —siempre y cuando el valor de los productos que elija no supere el valor de su producto —. Una vez se confirme la contraoferta, el usuario que realizó la primera oferta recibirá una notificación por correo.

El usuario que reciba una contraoferta podrá aceptar, rechazar o contraofertar nuevamente. En el caso de esta nueva contraoferta, el usuario solo podrá editar sus propios productos, es decir, hacer una re-elección de los productos que el otro usuario efectuó, no se podrá cambiar el producto en el que originalmente se estaba interesado.

2.3 Restricciones

- El sistema solo estará disponible para la plataforma Windows y solo funciona con sistemas de 64 bits.
- El uso de monedas o criptomonedas que sean externas a la aplicación, ejemplo: Peso Uruguayo, Euro, Bitcoin, Ethereum. está prohibido.
- Las UCUCoins son únicamente un valor de referencia para los artículos.
- Las transacciones sólo pueden efectuarse si los productos involucrados son iguales en valor de UCUCoins.
- El sistema no acepta cédulas que comiencen por cero[1].

2.4 Suposiciones

- Una vez que una transacción se confirma, no existen reembolsos.
- Los usuarios tendrán un nombre de usuario propio en el sistema.
- Todo producto que se publique en la app será mostrado en el feed principal.
- El usuario es libre de elegir el valor en UCUCoins de los artículos que publique.

[1] El siguiente link es útil para obtener cédulas uruguayas válidas: https://picandocodigo.github.io/ci_js/

2.5 Requisitos futuros

El desarrollo del sistema está orientado a ser extensible, es decir, que en un futuro existe la posibilidad de agregar muchas más funcionalidades o extenderse a diferentes plataformas, a continuación se mencionan algunas características que podrán ser agregadas al sistema y se consideran de gran utilidad en el mundo del software actual:

2.5.1 Chat entre usuarios

Los usuarios podrían enviar mensajes directos a modo de poder negociar de manera más efectiva y ponerse de acuerdo en algunas cuestiones como los valores de los productos a intercambiar, es necesario a veces tener una vía de comunicación instantánea. Como también se puede hacer un mal uso de esta herramienta, existirá la opción de reportar al usuario y bloquearlo para que no lleguen más mensajes de él en caso de que se considere que muestre una actitud que infrinja las normas de la comunidad. El caso sería estudiado por los administradores de sistema, los cuales decidirían si se sanciona al usuario y que sanción se aplicaría.

2.5.2 Aplicación multiplataforma

En base a la repercusión que tenga esta primera versión, se podría expandir el dominio de la aplicación en distintas plataformas, como por ejemplo: MacOS, Linux, Android, iOS, etc. Que la aplicación sea multiplataforma además de flexibilizar la forma de intercambio entre usuarios, es una buena forma de seguir expandiendo el negocio e incrementar presencia en posibles nichos de mercado.

2.5.3 Foro de consultas

De forma que los usuarios puedan seguir expandiendo la experiencia con la plataforma, la creación de un foro permitiría generar una red de colaboración colectiva, los usuarios harán publicaciones acerca de las consultas que tengan y otros usuarios podrán dar respuestas públicas acerca de las problemáticas planteadas. Existiría un moderador en el foro que revisará las publicaciones e intentará mantener las normas de la comunidad y un hilo de coherencia, por ejemplo, si las consultas ya fueron resueltas o la última respuesta fue hace más de diez días, las publicaciones serán catalogadas como Off-Topic, quedando inactivas para la respuesta pero archivadas por si en el futuro algún usuario necesita esa información.

3. Requerimientos

3.1 Requerimientos funcionales

A continuación, se pasarán describir los requerimientos funcionales los cuáles estarán divididos en módulos. Estos módulos son: Usuarios, Mercado, Feed y Notificaciones.

3.1.1 Módulo usuarios

Este módulo es el encargado de realizar las acciones correspondientes a la identificación y creación de usuarios con sus datos. Este módulo es importante ya que todo el sistema se regirá por los usuarios que lo forman y las transacciones que estos hacen.

Identificación del requerimiento	**RF_U01**
Nombre del Requerimiento	Alta de usuario
Descripción del requerimiento	Se podrá dar de alta a un nuevo usuario brindando: ● cédula ● nombre ● apellido ● teléfono ● correo ● nombre de usuario ● contraseña
Prioridad del requerimiento	Media

Identificación del requerimiento	**RF_U02**
Nombre del Requerimiento	Login de usuarios
Descripción del requerimiento	Ingreso al sistema para usuarios registrados utilizando su nombre de usuario y contraseña
Prioridad del requerimiento	Media

Identificación del requerimiento	**RF_U03**
Nombre del Requerimiento	Validación de cédula
Descripción del requerimiento	El sistema deberá validar que la cédula de un usuario que se intenta dar de alta es un cédula válida de la República Oriental del Uruguay
Prioridad del requerimiento	Alta

3.1.2 Módulo de mercado

Este módulo es inherente a las acciones de los usuarios necesarias para poder obtener el producto que desean e intercambiarlos por aquellos productos que tengan en posesión, plantea la razón principal de ser de este sistema.

Identificación del requerimiento	**RF_M01**
Nombre del Requerimiento	Alta de publicación
Descripción del requerimiento	Usuarios registrados podrán publicar sus artículos para el intercambio. Se deberán solicitar los siguientes datos del artículo: ● Nombre ● Descripción ● Valor en UCUCoins ● Imágenes del artículo
Prioridad del requerimiento	Alta

Identificación del requerimiento	**RF_M02**
Nombre del Requerimiento	Edición de publicación
Descripción del requerimiento	Las publicaciones activas podrán ser editadas por si es necesario realizar ajustes en los datos (nombre, descripción, valor) y/o editar las imágenes publicadas.
Prioridad del requerimiento	Media

Identificación del requerimiento	**RF_M03**
Nombre del Requerimiento	Realizar oferta
Descripción del requerimiento	Desde el feed, los usuarios podrán seleccionar una publicación sobre la cual realizar una oferta. Para concretar la oferta el usuario deberá elegir, de entre sus productos publicados, los que deseen incluir en el intercambio.
Prioridad del requerimiento	Alta

Identificación del requerimiento	**RF_M04**
Nombre del Requerimiento	Concretar trueque (Aceptar oferta)
Descripción del requerimiento	Cuando una oferta o contraoferta es aceptada, se deberá persistir la transacción y los usuarios involucrados recibirán los datos necesarios para ponerse en contacto. La notificación es por correo (mencionado en el módulo de notificaciones).
Prioridad del requerimiento	Alta

Identificación del requerimiento	**RF_M05**
Nombre del Requerimiento	Rechazar oferta
Descripción del requerimiento	El usuario que recibió una oferta la rechazó. Se cancela la transacción en curso y se le notifica al usuario que ejecutó la oferta. La notificación es por correo (mencionado en el módulo de notificaciones).
Prioridad del requerimiento	Alta

Identificación del requerimiento	**RF_M06**
Nombre del Requerimiento	Realizar contraoferta
Descripción del requerimiento	El usuario que recibió una oferta decide contraofertar. Se deberán desplegar las publicaciones del usuario que originalmente ejecutó la oferta y se tendrá libertad de elegir los productos que desee para incluir en la transacción, siempre y cuando el valor de los productos sea igual. Se le notificará al otro usuario que se le realizó una contraoferta. La notificación es por correo (mencionado en el módulo de notificaciones).
Prioridad del requerimiento	Alta

3.1.3 Módulo feed

Este módulo permitirá hacer conocer a los usuarios de manera intuitiva, los artículos disponibles para el correspondiente intercambio de bienes.

Identificación del requerimiento	**RF_F01**
Nombre del Requerimiento	Listado de publicaciones
Descripción del requerimiento	Se deberán listar todas las publicaciones activas de todos los usuarios en el feed principal de la aplicación.
Prioridad del requerimiento	Media

Identificación del requerimiento	**RF_F02**
Nombre del Requerimiento	Refrescar ofertas recibidas
Descripción del requerimiento	Cuando un usuario inicie sesión, se deberán obtener las ofertas recibidas y mostrarlas en el feed principal.
Prioridad del requerimiento	Media

3.1.4 Módulo de notificaciones

Este módulo en un principio será únicamente de notificaciones por correo. En este módulo se describirán los tipos de notificaciones que existirán.

Identificación del requerimiento	**RF_N01**
Nombre del Requerimiento	Notificación de oferta recibida
Descripción del requerimiento	Se notificará por correo cuando un usuario reciba una oferta sobre una de sus publicaciones.
Prioridad del requerimiento	Media

Identificación del requerimiento	**RF_N02**
Nombre del Requerimiento	Notificación de la aceptación de una oferta
Descripción del requerimiento	Los usuarios involucrados en la transacción recibirán los datos necesarios para ponerse en contacto: teléfono y correo electrónico.
Prioridad del requerimiento	Media

Identificación del requerimiento	**RF_N03**
Nombre del Requerimiento	Notificación sobre el rechazo de una oferta
Descripción del requerimiento	Notificar al usuario del rechazo de la oferta
Prioridad del requerimiento	Media

3.2 Requerimientos no funcionales

En esta sección se describirán los requerimientos no funcionales de UCUTrade, los cuales estarán ordenados por prioridad y se manejan 2 categorías de requerimientos:

- Requerimientos de la organización: aquellas decisiones tomadas por el grupo para el desarrollo

- Requerimientos del producto: aquellos aspectos que están dados por la letra del obligatorio

Identificación del requerimiento	**RNF_01**
Nombre del Requerimiento	Lenguaje de programación
Descripción del requerimiento	El sistema estará desarrollado en .Net 5 (C#)
Tipo de requerimiento	De la organización
Prioridad del requerimiento	Alta

Identificación del requerimiento	**RNF_02**
Nombre del Requerimiento	Framework de frontend
Descripción del requerimiento	Se utilizará el framework de windows forms para el diseño de la aplicación
Tipo de requerimiento	De la organización
Prioridad del requerimiento	Alta

Identificación del requerimiento	**RNF_03**
Nombre del Requerimiento	Motor de base de datos
Descripción del requerimiento	Como motor de base de datos, se utilizará SQL Server.
Tipo de requerimiento	De la organización
Prioridad del requerimiento	Media

Identificación del requerimiento	**RNF_04**
Nombre del Requerimiento	Servidor de base de datos
Descripción del requerimiento	El servidor de base de datos será una máquina virtual con sistema operativo Linux Ubuntu 20.04.
Tipo de requerimiento	Del producto
Prioridad del requerimiento	Media

4. Modelo entidad relación

En esta sección se podrán visualizar múltiples versiones del modelado realizado, con el fin de demostrar el proceso llevado a cabo y la evolución del modelo entidad-relación hasta llegar al modelo que efectivamente se utilizará en el sistema (versión 3).

Para modelar se utilizó la metodología ascendente debido a que se trata de una problemática desconocida para nosotros y se considera importante comenzar por identificar las entidades que formarán parte de la solución.

4.1 Versión 1

Esta versión fue la primer versión a la que se llegó intentando modelar el sistema, a continuación explicaremos las ideas principales, y las motivaciones que nos llevaron a realizar una nueva versión.

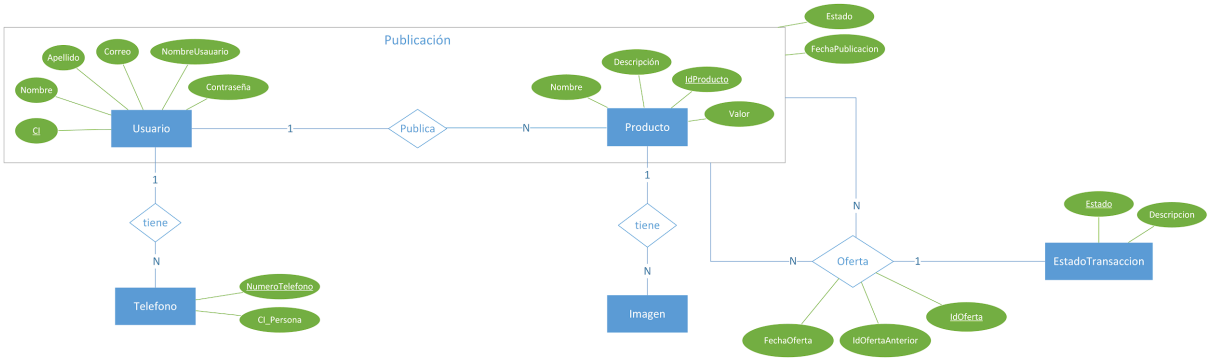

Figura 1 : MER-Versión 1

Las entidades identificadas en esta primera instancia son : Usuario, Telefono, Producto, Publicación, Imágen, Oferta, Estado de transacción.

Se comenzó modelando al **Usuario**, una entidad fundamental, ya que serán los actores principales del sistema, los cuales participarán en las publicaciones y transacciones. Estos tienen un nombre de usuario y contraseña además de los datos personales requeridos por el sistema — detallados en el documento de especificación de requerimientos —.

Posteriormente, se procedió a modelar la publicación. Se detectó que el **Producto** (o artículo) es aquello que la persona posee y desea publicar. Esta acción de publicar un producto da origen a la entidad **Publicación**, la cual surge gracias a la relación entre usuario y producto, además de tener una fecha de publicación la cual también es relevante para el sistema. Esto debe ser modelado como una agregación[2] ya que el fin es tratar a una relación entre 2 entidades como una nueva entidad.

[2] Se trata de un operador utilizado en el MER que transforma una relación en una entidad.

Por último se modelaron las ofertas. Estas se tratan de una relación entre publicaciones, es decir, una <u>auto-relación</u> de la entidad publicación. Que además tiene una fecha de realización, un identificador de oferta "anterior" la cual por defecto tendría un valor nulo y nos permitiría tener la trazabilidad de las contraofertas entre los usuarios y un estado que representaría si la oferta fue aceptada, rechazada o está pendiente de respuesta por parte del usuario destinatario.

A continuación se consideraron las combinaciones posibles de ofertas entre usuarios, lo cual nos ayudará a probar la robustez del modelo. Las combinaciones posibles son las siguientes:

Usuario 1			Usuario 2	
Publicacion1	100		Publicacion5	200
Publicacion2	150		Publicacion6	450
Publicacion3	100		Publicacion7	100
Publicacion4	500		Publicacion8	50

Usuario 1			Usuario 2	
Publicacion1	100		Publicacion5	200
Publicacion2	150		Publicacion6	450
Publicacion3	100		Publicacion7	100
Publicacion4	500		Publicacion8	50

Usuario 1			Usuario 2	
Publicacion1	100		Publicacion5	200
Publicacion2	150		Publicacion6	450
Publicacion3	100		Publicacion7	100
Publicacion4	500		Publicacion8	50

Usuario 1			Usuario 2	
Publicacion1	100		Publicacion5	200
Publicacion2	150		Publicacion6	450
Publicacion3	100		Publicacion7	100
Publicacion4	500		Publicacion8	50

Figura 2 : Combinaciones de ofertas entre usuarios

Como se aprecia en el modelo y en el ejemplo, la relación entre publicaciones de N a N. La solución planteada tiene el gran inconveniente de que los datos de una oferta deben repetirse si es que están involucradas varias publicaciones. Asimismo, no existe ninguna restricción a nivel de modelo que impida a un usuario realizar ofertas entre sus propias publicaciones. Un último detalle observado es que es muy complicado a través del modelo saber los roles de los usuarios involucrados en la oferta.

4.2 Versión 2

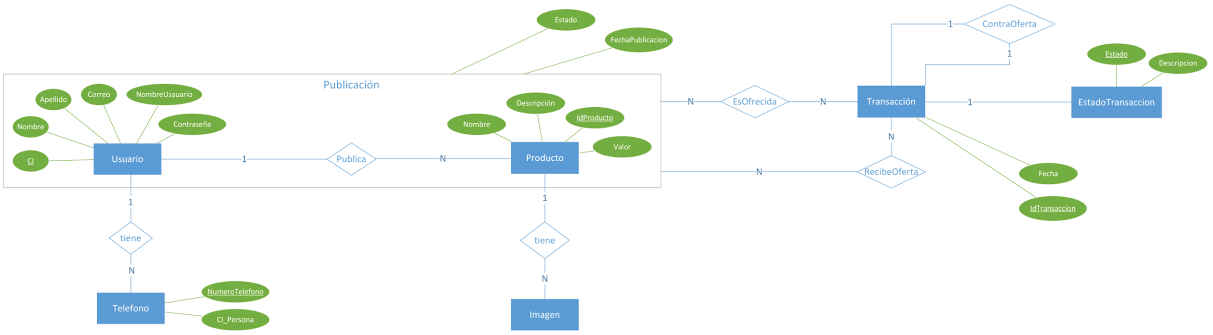

Figura 3 : MER-Versión 2

El objetivo principal de esta versión es solucionar los grandes problemas detectados en las ofertas de la versión anterior. Se parte de la idea de dejar de tratar a la oferta como una simple relación entre publicaciones, en esta versión aparece el concepto de **Transacción** esta entidad únicamente representa las ofertas existentes en el sistema sistema y el objetivo es relacionarla con las publicaciones, con el fin de no recurrir a la repetición de datos para "armar" una transacción por partes, como sucedía en el primer modelo. Además, se crearon dos relaciones entre Transacción y Publicación para poder representar cuáles son las publicaciones del usuario emisor y las del destinatario y así inferir sus roles dentro de la transacción.

Otra mejora es la existencia de una auto-relación entre transacciones para representar la **ContraOferta.** Esta auto-relación tiene como objetivo quitar el atributo IdOfertaAnterior a la transacción y llevar la trazabilidad en una tabla aparte.

Por más que esta alternativa solucione algunos problemas de la versión anterior, se siguen detectando problemas, principalmente no se restringe al usuario realizar ofertas entre sus propias publicaciones.

4.3 Versión 3

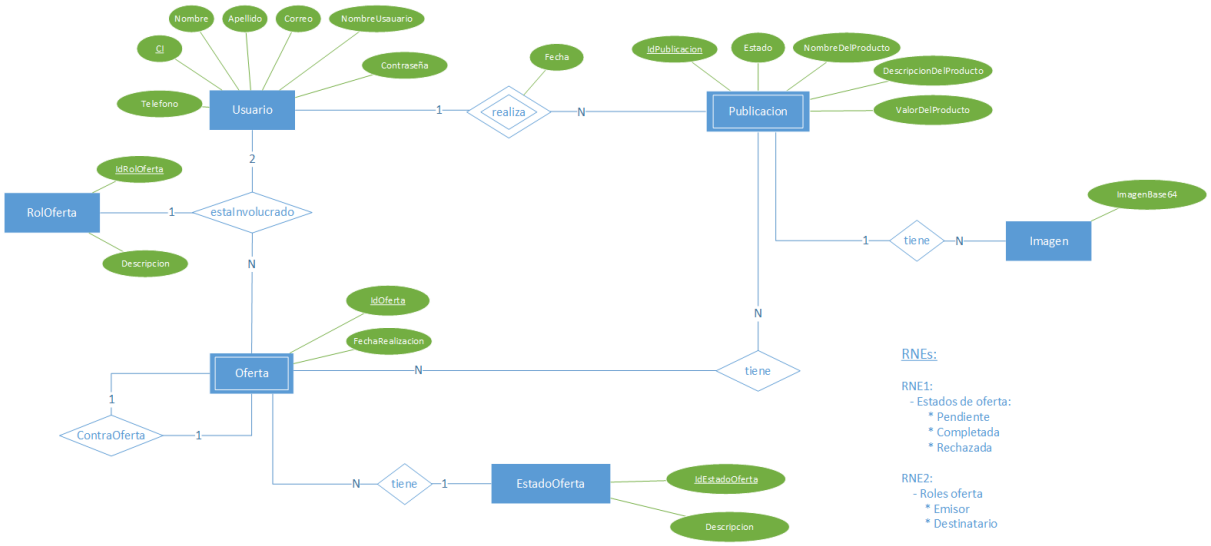

Figura 4 : MER-Versión 3

Como se observa, esta última versión contiene numerosos cambios. Se comenzó detectando que las entidades **Teléfono** y **Producto** carecen de sentido en el sistema. Por una parte, no es relevante guardar varios teléfonos de un usuario, por otra parte, en cuanto a los productos se considera que carecen de sentido ya que no es un sistema que requiera el registro de productos debido a que los mismos no existirían de no estar publicados, por lo tanto, los atributos de **Producto** fueron trasladados a **Publicación.** En este modelo existe una relación entre Usuario y Publicación para llevar el registro de las publicaciones realizadas.

El gran cambio se encuentra en la representación de las ofertas. Se parte del concepto de que una transacción es el acuerdo entre dos partes, en el ámbito de este sistema las partes son las personas, es decir, los usuarios del sistema.

Se considera entonces necesario relacionar a los Usuarios con las Ofertas. La naturaleza de esta relación es de "estar involucrado", ya que un usuario estará involucrado en varias ofertas, ya sea que estén activas, finalizadas o pendientes. Además, dentro de una oferta el usuario asume un rol, ya sea de emisor o destinatario, el cual en los modelos anteriores era un atributo inferido mediante otras relaciones.

En esta versión del modelo, se utiliza una entidad **RolOferta** la cual se asocia con Usuario y Oferta en la misma relación, de este modo es posible representar el rol que un usuario cumple en una oferta, impidiendo así que un usuario realice ofertas entre publicaciones propias, ya que podrá asumir un solo rol en la oferta.

De todos modos, es necesario mantener el registro de las publicaciones que los usuarios desean incluir en la oferta. Para ello debe también existir una relación entre las publicaciones y la oferta. Esta relación simplemente refleja las publicaciones que están incluidas en ella, ya sean del emisor o del destinatario. De este modo se logra "fortalecer" a la entidad usuario ya que es ella quién tiene la totalidad[3] sobre las publicaciones, es decir, que todas las publicaciones deben estar asociadas a un usuario y además estas no pueden "decidir" sobre los usuarios, como pasaba en las versiones anteriores que se debía inferir el rol del usuario en la oferta mediante las publicaciones involucradas en ella.

Estas dos relaciones mencionadas inevitablemente llevan a la aparición de un ciclo. Este ciclo está definido debido a que existen relaciones de distinta naturaleza entre las entidades involucradas. Por los requerimientos del sistema, es necesario llevar el registro de las publicaciones que realiza un usuario, por lo que romper esa relación sería inválido. Por otra parte, se debe llevar registro de las publicaciones que un usuario eligió incluir en una oferta determinada y teniendo en cuenta que una publicación puede ser incluida en ofertas distintas, debe existir relación entre oferta y publicación sin romper la relación de "propiedad" entre usuario y publicación. En cuanto a la relación Usuario-Oferta, podría ser eliminada si consideramos que dada una oferta, podríamos saber los usuarios involucrados en ella mediante las publicaciones que incluye. Eliminar esta relación llevaría inevitablemente a que vuelva a aparecer el problema de las versiones anteriores, o sea, que las publicaciones "decidan" el rol del usuario y como ya hemos dicho, que esto suceda no es correcto ya que inferir roles en la lógica de negocio mediante otra entidad que no sea el propio usuario no es adecuado.

Por último, vale la pena mencionar que el modelo es bastante auto explicativo en cuanto a la naturaleza de una oferta ya que si se observa la entidad a detalle, se aprecia que es una entidad claramente débil y que depende tanto de los usuarios que están involucrados y de las publicaciones que estos incluyen en ella, lo que nos lleva a concluir que el modelo presentado logra representar la realidad.

[3] Dadas 2 entidades relacionadas A y B, se dice que la relación es *total* con respecto a A si todos los elementos de A deben aparecer en alguna pareja de la relación.

4.4 Modelo lógico

Usuario (Ci, Nombre, Apellido, Correo, NombreUsuario, Contrasenia, Telefono)

Publicación (IdPublicacion, CiUsuario, FechaPublicacion, Estado, NombreProducto, DescripcionProducto, ValorProducto)

Imagen (IdPublicacion, ImagenBase64)

EstadoOferta (IdEstadoOferta, Descripcion)

Oferta (IdOferta, FechaRealizacion, EstadoOferta)

RolOferta (IdRolOferta, Descripcion)

UsuarioOferta (CiUsuario, IdOferta, IdRolOferta)

ContraOferta (IdOfertaNueva, IdOfertaAnterior)

PublicacionOferta (IdPublicacion, IdOferta)

5. Modelo funcional

Para diseñar el sistema se eligió un arquitectura en 3 capas:

- **D**ata **A**ccess **L**ayer (o capa de acceso a los datos)
- **B**usiness **L**ogic **L**ayer (o capa de lógica de negocio)
- **P**resentation **L**ayer (o capa de presentación)

Esta conocida arquitectura permite programar cada capa de forma independiente desacoplando los componentes con el fin de permitir la reutilización a futuro. Siendo la capa de acceso a los datos la más bajo nivel y presentación la de más alto nivel. Cada capa usa los servicios del nivel inmediatamente inferior y ofrece servicios a la capa inmediatamente superior dejando a la capa de negocio como intermediario entre los datos y el usuario final que se comunicará directamente como la capa de presentación.

5.1 Capa de acceso a datos

Esta capa cuenta con una clase llamada DatabaseContext que es la encargada de comunicarse directamente con la base de datos utilizando el conector de SQL Server. Esta clase tendrá una única instancia (patrón "***Singleton"***) en el sistema ya que se trata de una aplicación de escritorio que se comunicará directamente con una base de datos. Este aspecto lleva a que cada "cliente" de la aplicación tenga un único usuario de base de datos por lo que esta instancia mantendrá en memoria el connection string apropiado.

2.1.1 Patrón de diseño Repository

Los ***"repositories"*** son interfaces que encapsulan todas las operaciones que requiera **una entidad** dentro del negocio, por ejemplo: Insertar, Listar, Obtener por clave primaria, etc. Estas interfaces además serán las encargadas de crear las sentencias SQL y solicitar a DatabaseContext ejecutar la operación especificada por la implementación del método invocado. Cabe reforzar la idea de que únicamente se encargan de una sola tabla y no de sus relaciones con otras.

5.2 Capa de negocio

La capa de negocio es el ***"core"*** del sistema. Esta incluye los objetos que representan a las entidades del negocio y las clases encargadas de solicitar los datos a la capa inferior (provenientes de la base de datos) y brindarlos a presentación.

2.2.1 Services

Los ***"services"*** son interfaces un poco más complejas que los repositories, pero siguen el mismo objetivo de encapsular las operaciones de una entidad. Los services consumen al repositorio correspondiente a la misma entidad y además pueden requerir de comunicarse con otros services —además de con algún otro tipo de interfaz como puede llegar a ser un builder— .

El objetivo de los services es complementar las operaciones de un repository, ya que estos al tener la posibilidad de comunicarse con otros servicios indirectamente se podrán encargar de otras entidades con el fin de establecer las **relaciones** entre estas.

Ejemplo:

Se tiene la entidad Publicación que está relacionada con la entidad Usuario.

El negocio requiere de obtener los datos del usuario y de las publicaciones que éste ha realizado en un rango de fechas determinadas. Entonces, se contará con la interfaz *UserService* que se comunicará directamente con el *UserRepository* para obtener los datos del usuario deseado. A su vez, requerirá de comunicarse con *PostService* que tendría un método para obtener las publicaciones de un usuario determinado en el rango de fechas deseado, entonces el *UserService* podría crear una estructura que tenga un Usuario y el listado de sus publicaciones obtenidas

5.3 Capa de presentación

La capa de presentación es la encargada de desplegar la información solicitada por el usuario (obteniendola a través de la capa de negocio, es decir, consumiendo los services). En esta oportunidad esta capa será implementada utilizando Windows Forms.

5.4 Modelado de clases UML

A continuación se presenta el modelo de dominio y el modelado de repositories y services. Este último es un modelado genérico de las interfaces bases del patrón de diseño a utilizar.

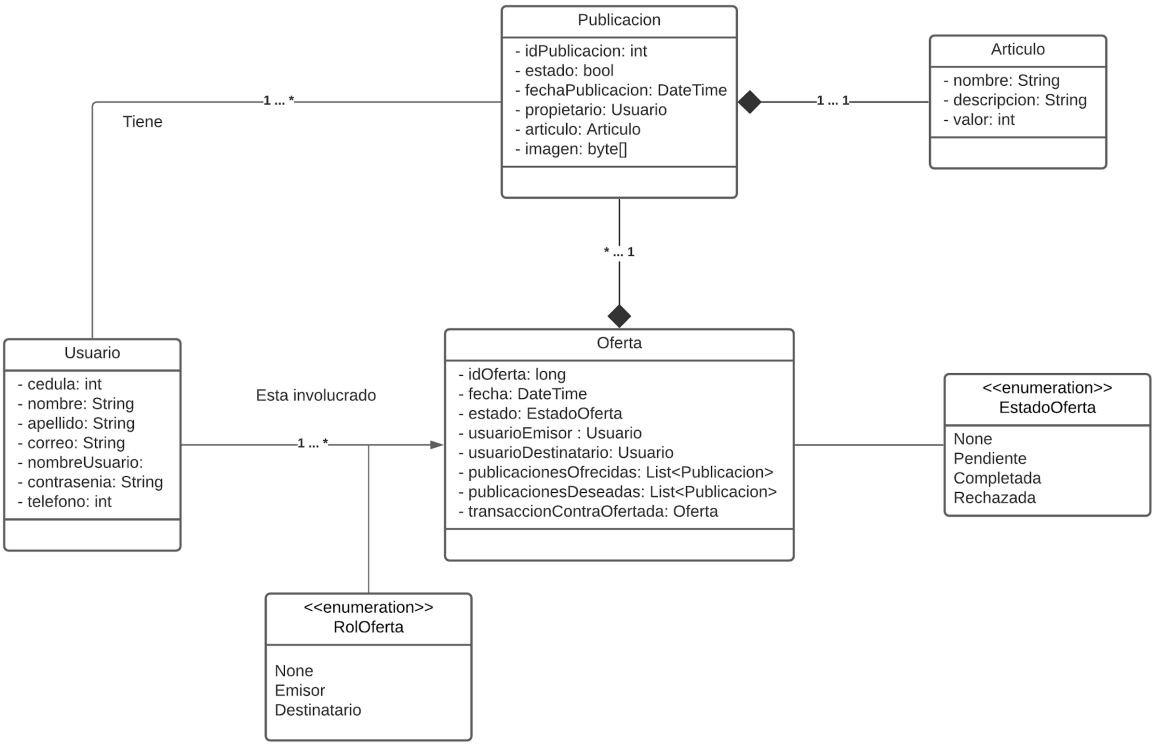

Figura 5 : Modelo de Domino

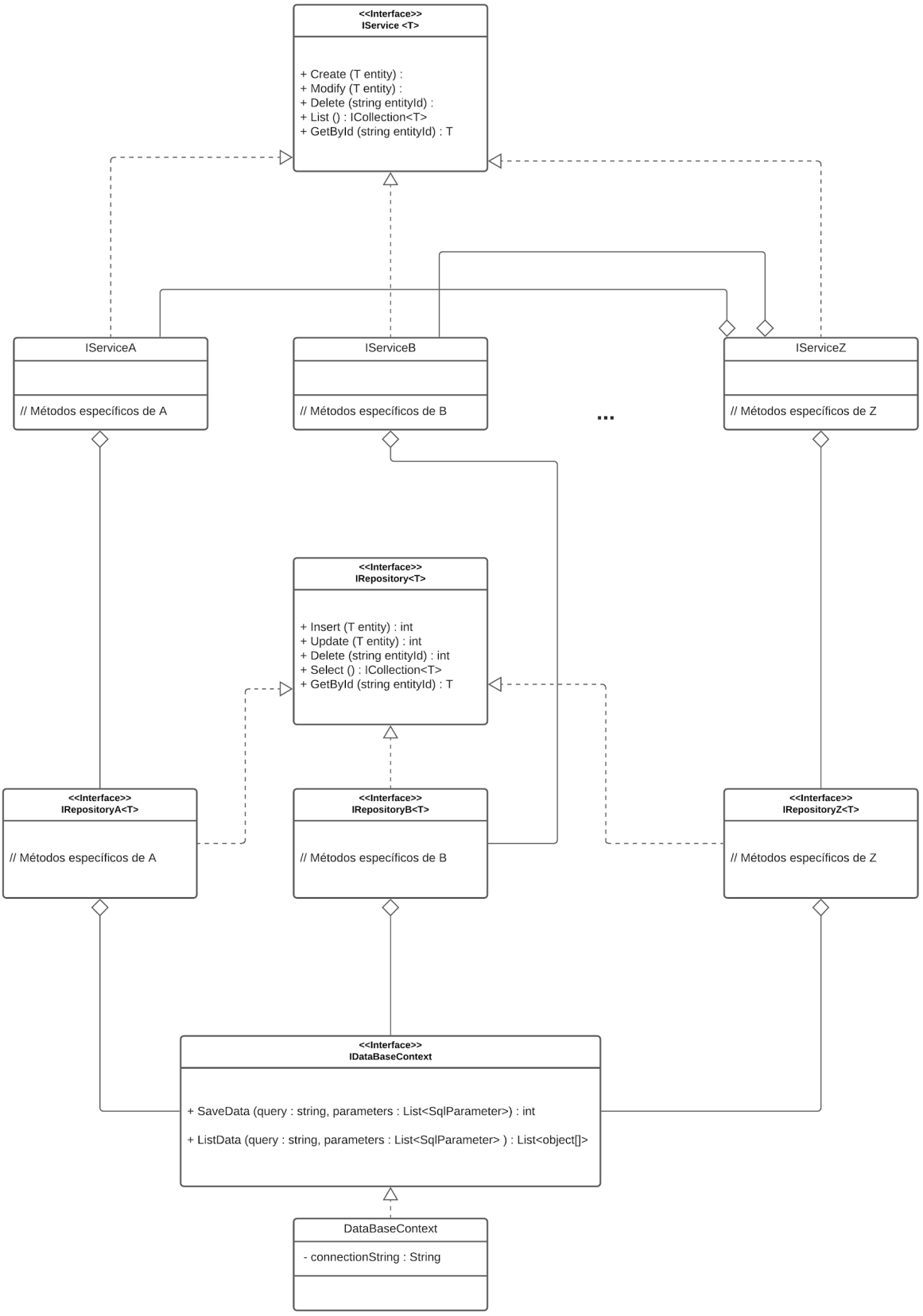

Figura 6 : Modelo genérico de DAL y Negocio

6. Bibliografía

— Quiroa, M. (2020, 6 febrero). *Transacción*. Economipedia. https://economipedia.com/definiciones/transaccion.html

— *Repository Design Pattern in C# with Examples*. (2021, 20 agosto). Dot Net Tutorials. Recuperado 19 de octubre de 2021, de https://dotnettutorials.net/lesson/repository-design-pattern-csharp/

— FING. (s. f.). *Pasaje de Modelo E-R a Modelo Relacional*. Fing.edu.uy. Recuperado 20 de octubre de 2021, de https://www.fing.edu.uy/tecnoinf/paysandu/cursos/2do/bd1/material/bd1-8.pdf

— FING. (s. f.-a). *Bases de Datos 1: Diseño Conceptual*. Fing.edu.uy. Recuperado 20 de octubre de 2021, de https://www.fing.edu.uy/tecnoinf/paysandu/cursos/2do/bd1/material/bd1-2.pdf

— Martinez López, Aura. (28 de mayo de 2019). Especificación de Requerimientos. https://topicosbasededatos997954789.files.wordpress.com/2019/06/copia-de-ejemplo-formato-ieee-830.pdf

— Downey, Lucas. (31 de agosto de 2021). Local Exchange Trading Systems. https://www.investopedia.com/terms/l/local-exchange-trading-systems-lets.asp

— Autor Desconocido. (4 de julio de 2020). Sistema de cambio local. https://es.wikipedia.org/wiki/Sistema_de_cambio_local

— Miguel Yasuyuki Hirota. (10 de Junio de 2014). LETS: una moneda independiente del euro. https://elpais.com/elpais/2014/06/10/alterconsumismo/1402388311_140238.html