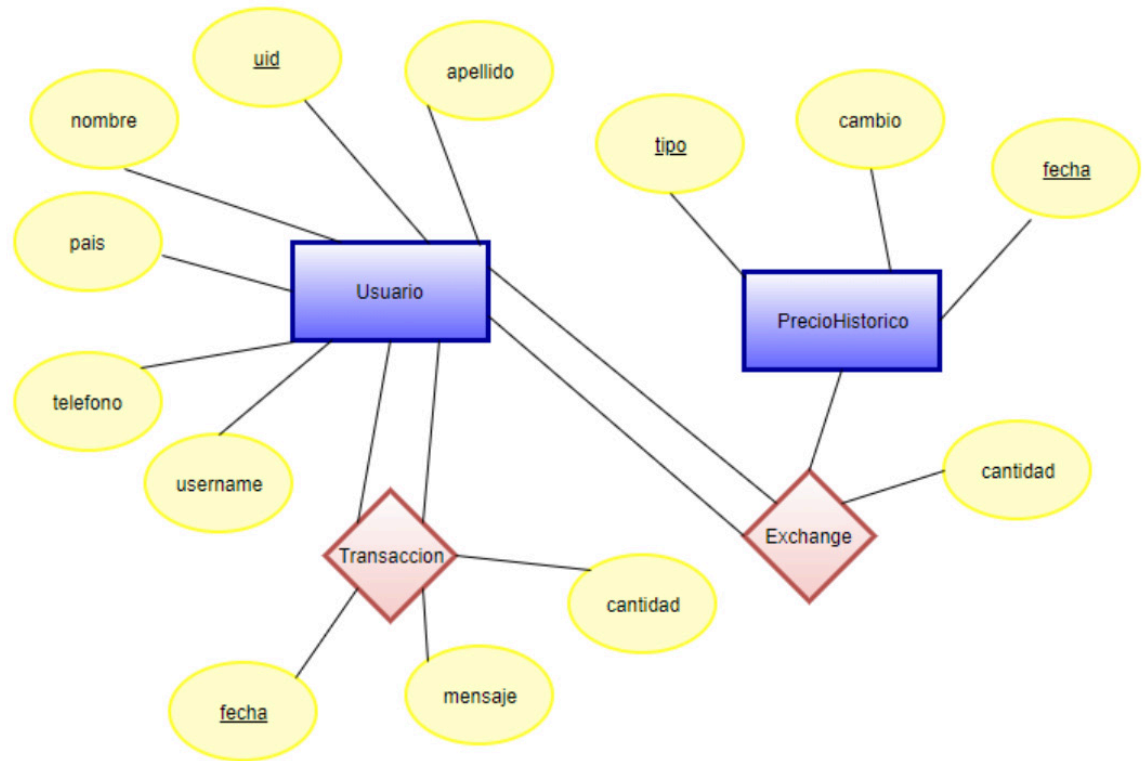


# Diseño de Esquema 2.0 Grupo 1

## Diagrama entidad/relación

Ignacio Acosta  
Manuel Valdivia



El Usuario es una entidad que representa una persona que puede hacer transacciones para enviar zorcales a otro usuario. Además debe hacer un exchange con otro usuario para poder recibir zorcales por primera vez.

La Transacción es una relación que representa el intercambio de zorcales entre usuarios.

El PrecioHistorico es una entidad que representa el valor de los zorcales en pesos chilenos o en dólares, según el tipo, en una fecha en específico.

Por último, el Exchange es una relación entre dos usuarios y los precios históricos que representa la cantidad de plata en dolares, o pesos chilenos, según el tipo, que depositó un usuario a otro. En nuestro modelaje supusimos que existe un usuario con id = 0, que empieza con una gran cantidad de zorcales que es a quien se le pueden comprar zorcales a través de un exchange.

## Descripción de las tablas

Por los algoritmos aprendidos en clases, se generan cuatro tablas: Usuario, PrecioHistórico, Transacción y Exchange. Por el mismo algoritmo, los atributos de Usuario y PrecioHistórico son los mismos que en los diagramas, respetando sus llaves primarias (las subrayadas). Y para las relaciones Transacción

y Exchange ocurre exactamente lo mismo, pero se le agregan los atributos, como parte de las llaves primarias, from\_id, to\_id y lo mismo más fecha y tipo respectivamente.

Nos quedan las siguientes relaciones:

Usuario	<b>uid</b> <i>int</i>	<b>nombre</b> <i>varchar(30)</i>	<b>apellido</b> <i>varchar(30)</i>	<b>email</b> <i>varchar(30)</i>	<b>telefono</b> <i>varchar(30)</i>	<b>país</b> <i>varchar(30)</i>
	Identificador único de cada usuario	Nombre real del usuario	Apellido real del usuario	Correo personal del usuario	Teléfono del usuario	País de origen del usuario
Transaccion	<b>from_id</b> <i>int</i>	<b>to_id</b> <i>int</i>	<b>fecha</b> <i>timestamp</i>	<b>mensaje</b> <i>varchar(255)</i>	<b>cantidad</b> <i>float</i>	
	ID del usuario que hace la transacción	ID del usuario que recibe la transacción	Fecha de la transacción	Mensaje adjunto de la transacción	Cantidad de Zorsales de la transacción	
PrecioHistorico	<b>fecha</b> <i>date</i>	<b>tipo</b> <i>varchar(10)</i>	<b>cambio</b> <i>float</i>			
	Fecha del tipo de cambio	Tipo de moneda del cambio.	Equivalencia en el tipo mencionado			
Exchange	<b>from_id</b> <i>int</i>	<b>to_id</b> <i>int</i>	<b>fecha</b> <i>timestamp</i>	<b>tipo</b> <i>varchar(10)</i>	<b>cantidad</b> <i>float</i>	
	ID del usuario que hace la transacción	ID del usuario que recibe la transacción	Fecha de la transacción	Tipo de moneda de la transacción	Cantidad de dinero de la transacción	

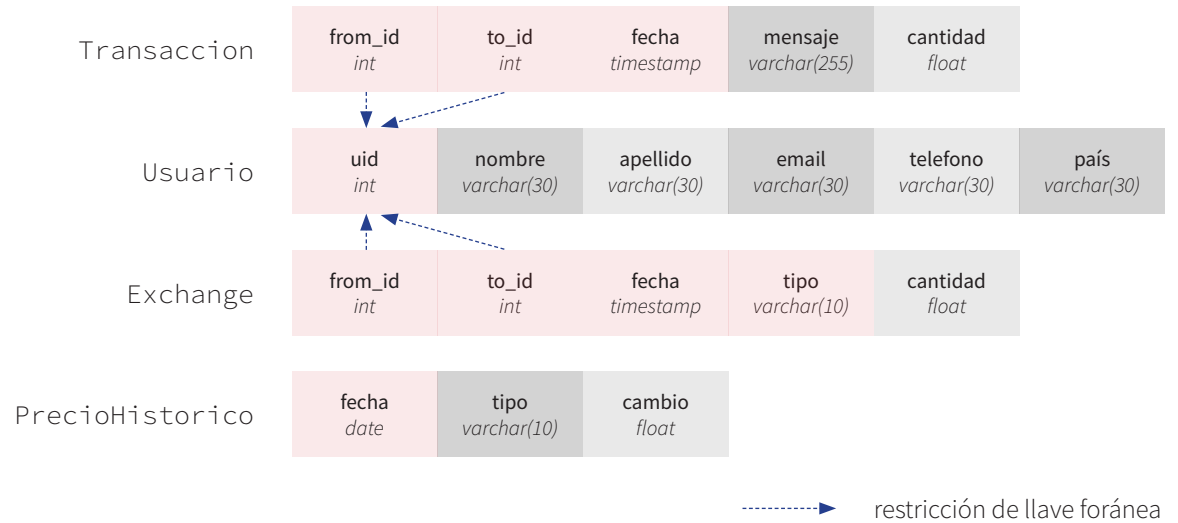
llave primaria

## Dependencias funcionales de las tablas

Usuario	uid → username, teléfono, país, nombre, apellido
Transaccion	to_id, fecha, from_id → mensaje, cantidad
PrecioHistorico	tipo, fecha → cambio
Exchange	to_id, fecha, tipo, from_id → cantidad

Como la parte izquierda de todas las dependencias son llaves primarias minimales, simplemente se procede con el algoritmo y cada una de estas dependencias se convierte en una relación, con la parte de la izquierda como llaves primarias y la parte derecha como atributos normales.

# Llaves foraneas y cambio respecto a Entrega 1



Tenemos cuatro llaves foráneas. Los to\_id y los from\_id de Transacción y Exchange que tienen como padre uid de Usuario, esto para que solo se puedan hacer Transacciones y Exchange entre Usuarios ya existentes.

Como se mencionó en la generación del esquema final: como la parte izquierda de todas las dependencias son llaves primarias minimales, simplemente se procede con el algoritmo y cada una de estas dependencias se convierte en una relación. De esta forma, todas son BCNF porque la parte izquierda de las dependencias son llaves primarias, lo mismo dicho por la definición.

En comparación el esquema de la entrega 1, la relación Usuario solo se le cambió el atributo username por email (gusto personal). En la relación

Transacción eliminamos la llave tid ya que con los atributos from\_id, to\_id y fecha ya teníamos una llave primaria. Para precio histórico eliminamos los atributos tipo\_usd y tipo\_clp porque para realizar ciertas consultas hubiésemos necesitado un "if". Por esto, decidimos agregar los atributos cantidad y tipo.

Por último, en la relación Exchange eliminamos exid por la misma razón que en la relación Transacción e hicimos de la llave los atributos from\_id, to\_id y fecha.

Además para todas las fechas decidimos cambiar de date a timestamp, y agregamos una llave foránea a Exchange y a Transacción haciendo referencia a User para que las transacciones y exchanges solo se puedan hacer entre usuarios.

## Consultas SQL

1. Dado un usuario (\$uid) y un día (\$fecha), liste todas las transacciones de un usuario en ese día

```
SELECT Remitentes.nombre, Remitentes.apellido, Destinatarios.
nombre, Destinatarios.apellido, cantidad, mensaje
FROM Transaccion, Usuario AS Remitentes, Usuario AS Destinatarios
WHERE Remitentes.uid = Transaccion.from_id
AND Destinatarios.uid = Transaccion.to_id
AND fecha::date = '$fecha'
AND from_id = $uid;
```

2. Dado un día (\$fecha) y una cantidad de zorzales (\$zorzales), entregue la equivalencia en pesos y en dólares de esa moneda.

```
SELECT tipo, cambio, cambio * $zorzales AS conversion
FROM PrecioHistorico
WHERE PrecioHistorico.fecha::date = '$fecha';
```

3. Dado un usuario (\$uid), muestre su última transacción

```
SELECT Remitentes.nombre, Remitentes.apellido, Destinatarios.
nombre, Destinatarios.apellido, fecha, cantidad, mensaje
FROM Transaccion, Usuario AS Remitentes, Usuario AS Destinatarios
WHERE Remitentes.uid = Transaccion.from_id
AND Destinatarios.uid = Transaccion.to_id
AND from_id = $uid
ORDER BY fecha DESC LIMIT 1;
```

4. Liste todos los usuarios de un cierto país (\$pais) de procedencia

```
SELECT *
FROM Usuario
WHERE pais LIKE '%$pais%';
```

5. Muestre el precio promedio de los zorzales en el mes pasado

```
SELECT tipo, AVG(cambio)
FROM PrecioHistorico
WHERE (EXTRACT(MONTH FROM fecha) + 1 = EXTRACT(MONTH FROM CURRENT_
TIMESTAMP))
GROUP BY tipo;
```

6. Indique cuál día del mes pasado recibió la mayor cantidad de transacciones, y en cuál día se transó mayor cantidad de zorzales

```
SELECT *
FROM (SELECT COUNT(cantidad) AS max_qty,
      Transaccion.fecha::date AS fecha_max_qty
      FROM transaccion
      GROUP BY Transaccion.fecha::date
      ORDER BY COUNT(cantidad) DESC LIMIT 1) AS A,
      (SELECT SUM(cantidad) AS max_amount,
      Transaccion.fecha::date AS fecha_max_amount
      FROM transaccion
      GROUP BY Transaccion.fecha::date
      ORDER BY SUM(cantidad) DESC LIMIT 1) AS B;
```

7. Dado un usuario (\$uid), calcule la cantidad de zorzales que posee

```
SELECT A.ingresos, B.gastos, C.inicial,  
       A.ingresos + B.gastos + C.inicial AS balance  
FROM (SELECT COALESCE(SUM(cantidad), 0) AS ingresos  
      FROM Transaccion  
      WHERE to_id = $uid) AS A,  
      (SELECT COALESCE(SUM(-cantidad), 0) AS gastos  
      FROM Transaccion  
      WHERE from_id = $uid) AS B,  
      (SELECT COALESCE(SUM(Exchange.cantidad / PrecioHistorico.cambio), 0)  
      AS inicial  
      FROM Exchange, PrecioHistorico  
      WHERE Exchange.fecha::date = PrecioHistorico.fecha  
      AND Exchange.tipo = PrecioHistorico.tipo  
      AND Exchange.from_id = $uid) AS C;
```