

**INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y CIENCIAS SOCIALES Y ADMINISTRATIVAS (UPIICSA)**

Ingeniería en Informática

**Ingeniería del Conocimiento**

**Banco Pípila**

**Profesor: Ramírez Flores Gilberto Antonio**

**INTEGRANTES:**

* Cordero Hernández Pablo Cesar 2015600437
* Osorio Cuello Edgar Hiram 2016602330
* Pastrana Jiménez Brenda Sulem 2016602726

Secuencia: 3NM7

Iztacalco, CDMX a 13 de septiembre de 2019

**¿Se puede reconstruir todas las transacciones que se han hecho con un billete?**

Hasta el momento en Banco Pípila no se puede hacer ya que no existe una tabla que nos permita guardar un historial, además de que las monedas no tienen número de serie como para poder identificarlas.

|  |  |  |
| --- | --- | --- |
| Idmon | Valor | Tipo |
| qwertyui4212 | 2 | moneda |
| dldaiwl45556 | 1 | moneda |
| Nvmxjo55456 | 50 | billete |
| Sjadasd87546 | 20 | billete |

Además es poco factible determinar en que transacciones se han usado ciertas monedas debido a que se puede usar más de una del mismo valor, por lo cual haría una tabla muy extensa tanto en campo como en datos, esto a su vez reduciría el poder de procesamiento y limitando otro tipo de transacciones más importantes.

**Algoritmo de conversión de correo-número de cuenta**

Hasta el momento la pagina solo tiene declaradas etiquetas input que envían los datos capturados a una tabla designada como usuarios donde se registran los datos capturados.



La asignación de número de cuenta se realiza gracias a userId, este usuario es agregado a la base de datos de la siguiente manera:

private void writeNewUser(String userId, String name, String email) {

User user = new User(name, email);  
  
    mDatabase.child("users").child(userId) .setValue(user);  
}

**Algoritmo de construcción del número de transacción  
¿Usa fecha, hora, emisor y receptor?**

No usa ninguno de los anteriores, usa un ID de transacción designado por la plataforma de Firebase

private void onStarClicked(DatabaseReference postRef) {  
    postRef.runTransaction(new Transaction.Handler() {  
        @Override  
        public Transaction.Result doTransaction(MutableData mutableData) {  
            Post p = mutableData.getValue(Post.class);  
            if (p == null) {  
                return Transaction.success(mutableData);  
            }  
            if (p.stars.containsKey(getUid())) {  
                // Unstar the post and remove self from stars  
                p.starCount = p.starCount - 1;  
                p.stars.remove(getUid());  
            } else {  
                // Star the post and add self to stars  
                p.starCount = p.starCount + 1;  
                p.stars.put(getUid(), true);  
            }  
  
            // Set value and report transaction success  
            mutableData.setValue(p);  
            return Transaction.success(mutableData);  
        }  
        @Override  
        public void onComplete(DatabaseError databaseError, boolean b,  
                               DataSnapshot dataSnapshot) {  
            // Transaction completed  
            Log.d(TAG, "postTransaction:onComplete:" + databaseError);  
        }  
    });  
}