Diseños

Diseño de la Base de Datos

MySql

Debido a sus características relacionales, la base de datos MySql es ideal para la solución de este problema. La base de datos está construida de la siguiente manera:

Tabla *persona*

En esta tabla aparecen los datos principales de las personas añadidas a la base de datos. Entre estos datos se encuentran:

- Nombre.
- Apellidos.
- Cédula.
- Tipo de sangre (A, B, AB, O).
- Rh de la sangre (positivo o negativo).
- Usuario de *Telegram*.
- Fecha de la última donación.

Tabla *sedeBanco*

En esta tabla se encuentran solamente las sedes del banco de sangre. Entre los datos de esta se encuentran:

- Nombre de la sede u hospital sede.
- Ubicación geográfica de la misma.

Tabla *donacion*

En esta tabla se encuentran los datos de una donación de una persona, efectuada en cierta sede u hospital, aca se encuentran llaves foráneas a otras tablas. Entre los datos de esta tabla se encuentran:

- Llave foránea a la persona donadora.
- Llave foránea a la sede receptora y almacenadora de dicha donación.
- Fecha de donación.
- Bandera la cual me dice si dicha sangre aún se encuentra en stock o si ya fue sujeto de una transfusión.

Tabla *listaEspera*

En esta tabla se encuentran los datos de una persona que se encuentra esperando para recibir una tranfución de sangre en cierto banco de sangre u hospital, aca se encuentran llaves foráneas a otras tablas. Entre los datos de esta tabla se encuentran:

- Llave foránea a la persona receptora.
- Llave foránea a la sede encargada de administrar dicha transfusión.
- Fecha de inscripción en lista de espera.

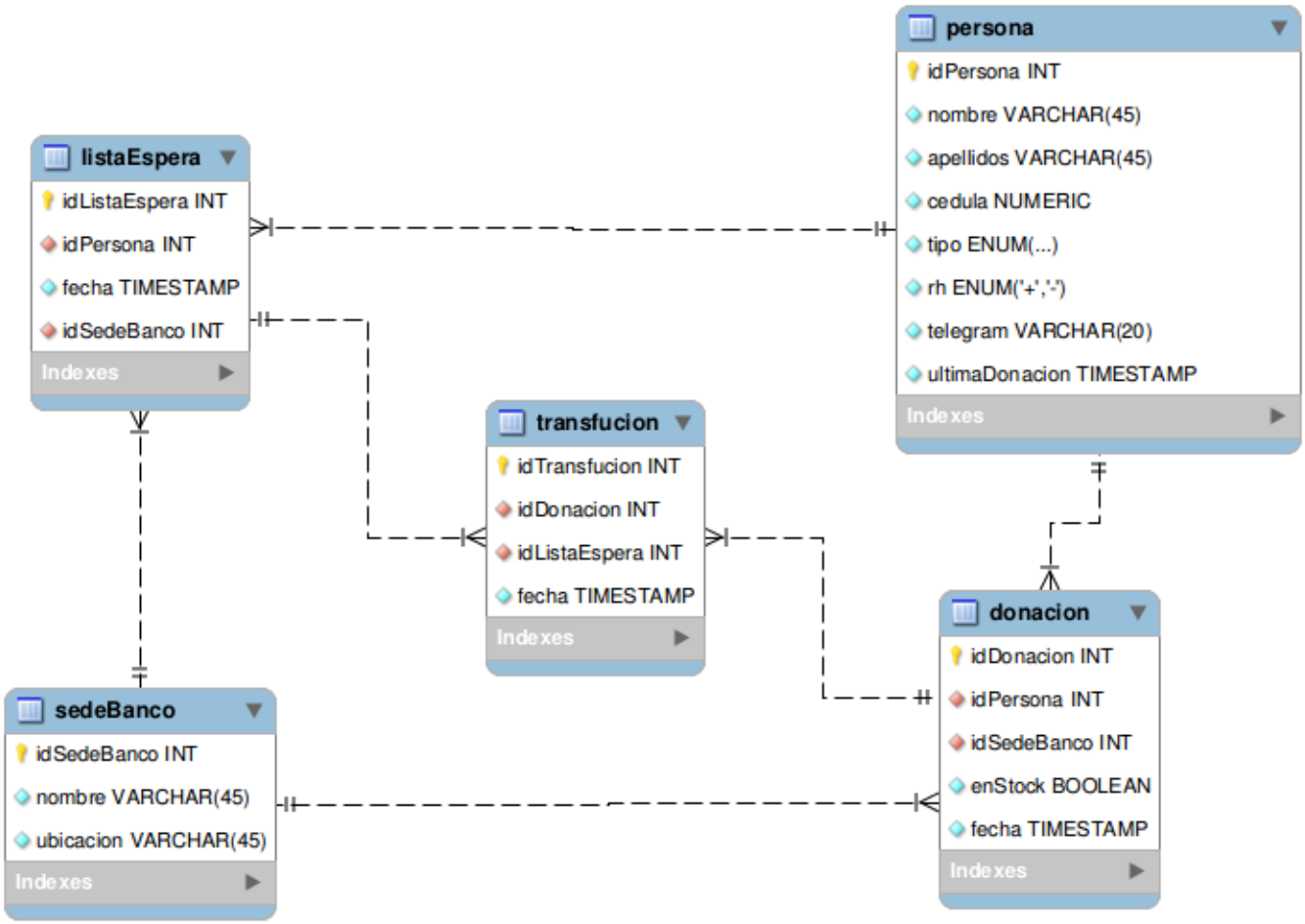
Tabla *transfucion*

En esta tabla se encuentran los datos de una transfusión de sangre, efectuada en cierta sede u hospital, aca se encuentran llaves foráneas a otras tablas. Entre los datos de esta tabla se encuentran:

- Identificador de la donación y sus datos.
- Identificador de la lista de espera y sus datos.
- Fecha de la transfusión.

NOTA: Todas las tablas tienen sus llaves primarias identificadoras, las cuales son únicas y autoincrementables.

Imagen del diseño de la base de datos



MongoDB

Usando *MongoDB* almacenaremos los mensajes que se transmitirán a la población en general, esto ya sea vía *Telegram* o *Twitter*. Debido a que *MongoDB* almacena colecciones en formato *JSON*, se crearán 2 colecciones, una para los mensajes y la otra para manejar los traslados de stock entre las

sedes u hospitales.

Creación de la DB

Utilizamos los siguientes comandos para crear la base de satos y las collecciones:

```
> db.dropDatabase()
{ "ok" : 1 }
> use basesii
switched to db basesii
> db.createCollection("mensajes")
{ "ok" : 1 }
> db.createCollection("traslados")
{ "ok" : 1 }
>
```

Colección *Mensajes*

En esta collección almacenaremos el *JSON* de los mensajes de la siguiente forma:

```
{
    medio: "Telegram",
    receptor: ["@Tavog14","@kmoragas","@quamtics"],
    mensaje: "Buenos dias, se le comunica que el Hospital de Alajuela
ocupa recervas de sangre tipo A+, acerquese a donar y ayudenos a salvar
vidas",
    fecha: new Date(2017,04,11,16,05)
}
```

A continuación se muentran los datos y su respectiva información:

- Medio: medio de comunicación, ya sea *Telegram* o *Twitter*.
- Receptor: quienes reciben el mensaje (*Telegram*) o el dueño de la cuenta de *Twitter*.
- Mensaje: el mensaje que se envia a los receptores.
- fecha: fecha en la que fue emitido el mensaje.

Colección *Traslados*

En esta collección almacenaremos el *JSON* de los traslados de la siguiente forma:

```
{
    fecha: new Date(2013,11,10,2,35),
    idDonacion: 126,
    sede_emisora: "Hospital de Heredia",
    sede_receptora: "Hopital Nacional de Niños"
}
```

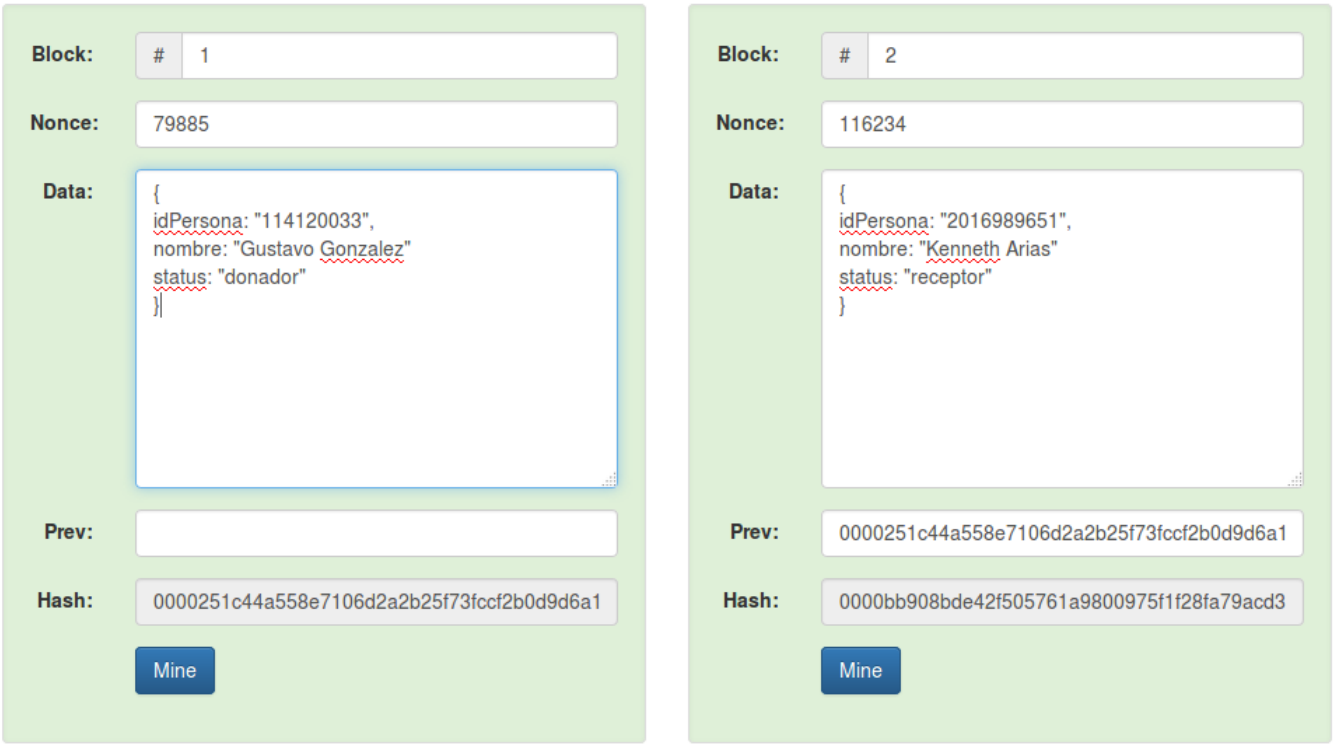
A continuación se muentran los datos y su respectiva información:

- fecha: fecha del traslado de la sangre.
- idDonacion: llave primaria en la tabla de *MySql donacion*, esto para no perder el dato.
- sede_emisora: sede u hospital de la cual fue extraida la muestra de sangre.
- sede_receptora: sede u hospital a la cual fue trasladada la muestra de sangre.

NOTA: en *MongoDB* los id son autogenerables.

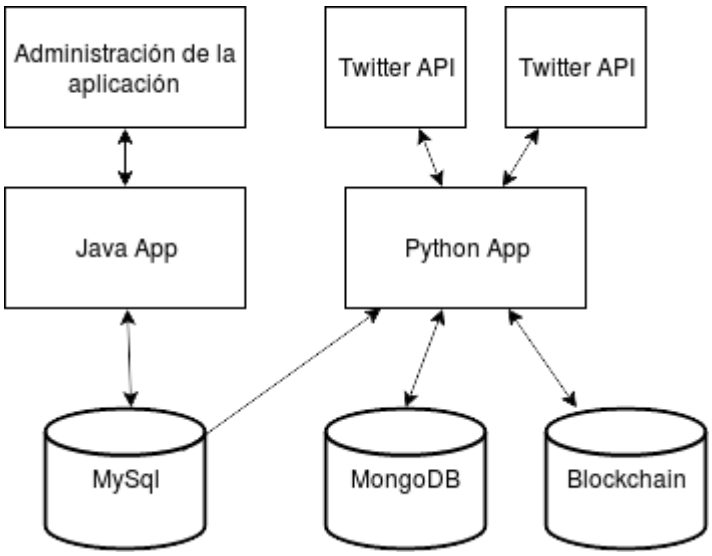
Blockchain

Para la base de datos basada en *Blockchain* usaremos(por definir). En sí, lo que se tiene pensado es en un bloque el cual tiene la información referente a un donador, toda esta información escrita en un formato *JSON* y en un segundo bloque de información, los datos del receptor. Estas asociadas con su respectivo *nonce* y su respectivo *hash*. Cada insert se vera de la siguiente manera:



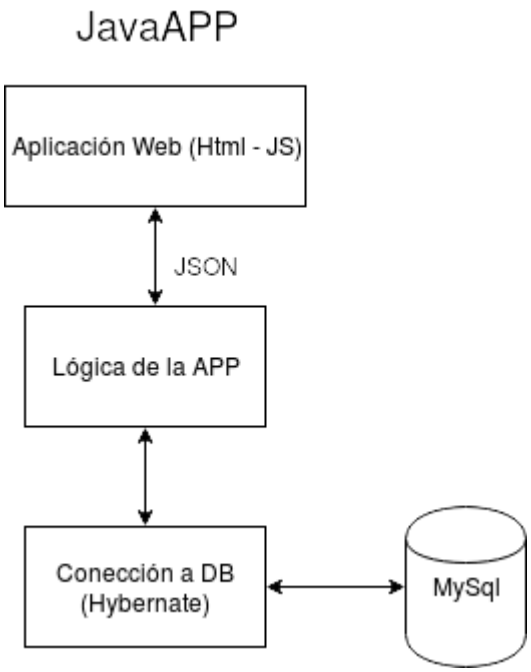
Diseño de la arquitectura de software

El diseño de la arquitectura general contempla el uso de 3 bases de datos (*MySql, MongoDB, Blockchain(por definir)*), 2 lenguajes de programación (*Java, Python*). La siguiente imagen ejemplifica lo anterior:



Java APP

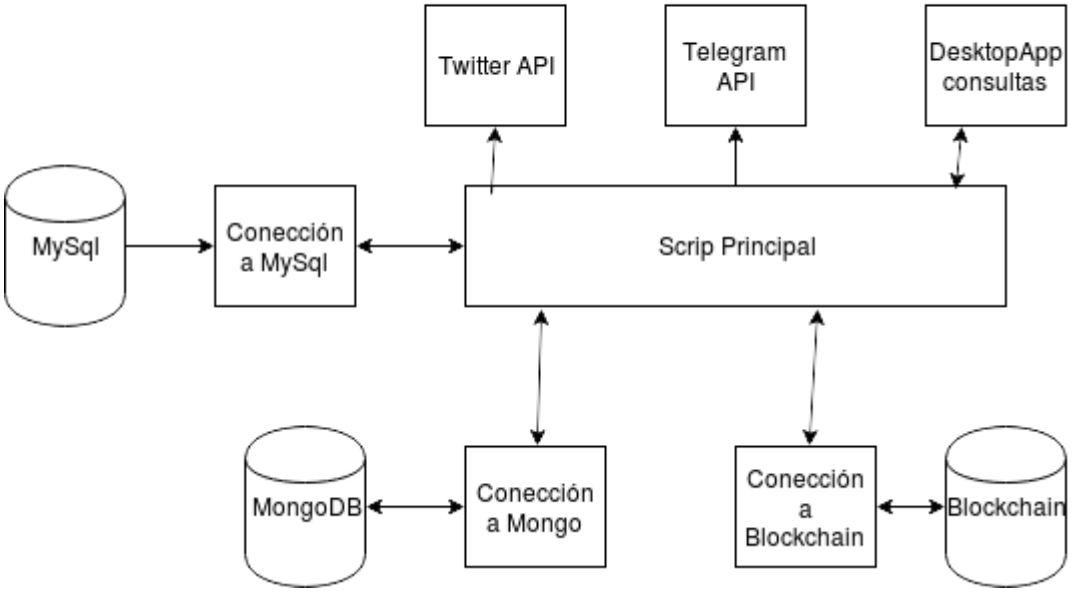
Para la aplicación *Java*, tendremos un diseño de capas, en esta interactúa con una interfaz *html* usando la tecnología *RESTful* a travez de *JSON*. Intermedio tendremos una capa de lógica de la aplicación y más abajo una capa de acceso a datos usando *Hybernate* usando ya sea *DAOs* o el patrón de diseño *Repository*. Para toda esta parte posiblemente se usará de algún framework, seguramente *Spring* para sacar al máximo las ventajas del modelo *MVC*. La siguiente imagen ejemplifica lo anterior:



Python APP

Para la aplicación *Python*, tendremos tambien un diseño de capas, pero esta vez los datos en vez de ser recibidos del usuario, serán recibidos de la base *MySql* y la aplicación actua como mediador entre estos datos y las APIs de *Twitter* y *Telegram* para luego ser almacenadas en las bases de datos *MongoDB* y *Blockchain*. La siguiente imagen ejemplifica lo anterior:

PythonAPP


Twitter API
Telegram API
DesktopApp consultas
MySql
Conección a MySql
Scrip Principal
MongoDB
Conección a Mongo
Conección a Blockchain
Blockchain