

# Sistemas de Información para Internet

## Tarea 1: Modelo Entidad-Relación y Entidades JPA



### Integrantes del equipo grupo C:

- Iván García Aguilar (Coordinador de Grupo).
- Antonio Manuel Burgueño Romero.
- Adrián Román Navas.
- Miguel Guerrero Ramírez.

## Documento redactado por:

-Adrián Román Navas.

-Iván García Aguilar.

## -Introducción:

En esta entrega, se ha definido el modelo de Entidad-Relación en base a los requisitos documentados y validados por el usuario.

Dicho modelo se ha realizado haciendo uso de la herramienta MagicDraw.

En este documento, se especificará como a partir del modelo propuesto, es posible cumplir todos los requisitos propuestos.

Una vez presentado el modelo con su respectiva explicación, a continuación, se adjuntará imágenes de cada una de las clases JPA definidas.

Toda la documentación y código que vayamos generando, se almacenará en un repositorio de Github creado para el proyecto.

El link para el acceso al mismo es el siguiente:

[https://github.com/IvanGarcia7/SII-ACOES-HEFESTO\\_SOFTWARE](https://github.com/IvanGarcia7/SII-ACOES-HEFESTO_SOFTWARE)

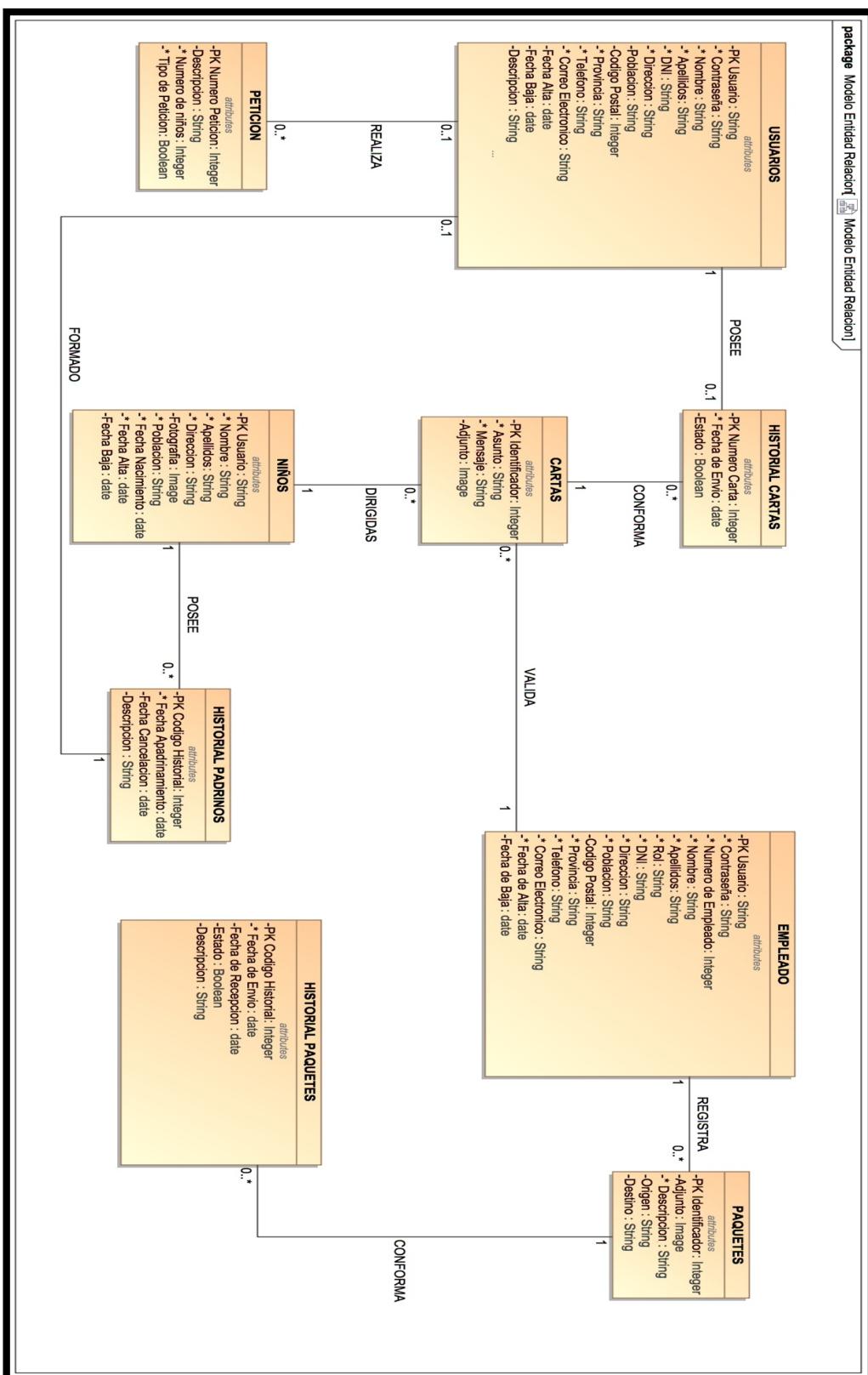
### Último commit realizado:



En el archivo README.md se puede visualizar información acerca de la estructura y del contenido almacenado en el repositorio.

Una vez definida una breve introducción, pasamos a desarrollar cada una de las partes propuestas:

# -Modelo Entidad - Relación:



Para una correcta visualización del mismo, se ha rotado la imagen, no obstante, dicho modelo también se ha subido al repositorio de trabajo.

## EXPLICACIÓN DEL MODELO ENTIDAD RELACIÓN.

Hemos decidido diseñar un modelo entidad-relación, atendiendo a los criterios proporcionados por el cliente, que consta de 9 entidades, las cuales son:

- USUARIOS-
- PETICIÓN-
- HISTORIAL CARTAS-
- CARTAS-
- NIÑOS-
- HISTORIAL PADRINOS-
- EMPLEADO-
- HISTORIAL PAQUETES
- PAQUETES-

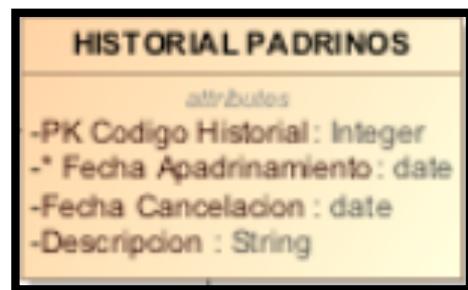
Procederemos a continuación a explicar cada entidad:

### HISTORIAL PADRINOS

La presente entidad corresponde con HISTORIAL PADRINOS, la cual almacenará los siguientes atributos:

- Código Historial: será la PK de la entidad con la que identificaremos de forma inequívoca un historial.
- Fecha de apadrinamiento.
- Fecha cancelación.
- Descripción: incluiremos una breve descripción del historial.

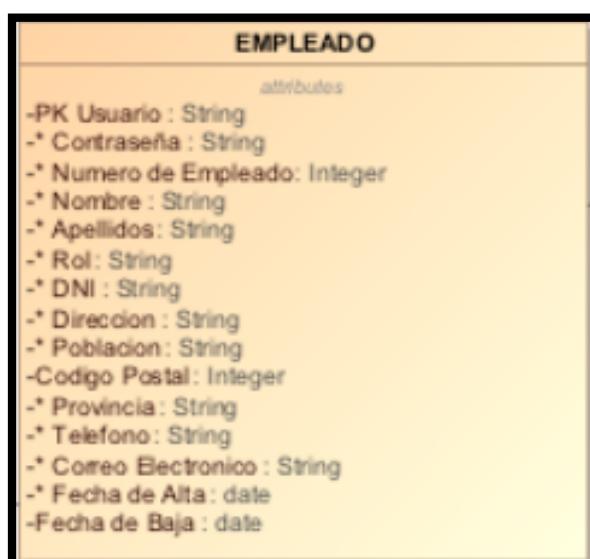
Esta entidad se realizó con el fin de almacenar un conjunto de históricos que faciliten información sobre la relación entre usuarios y niños. Gracias a esta entidad, un empleado puede acceder a los datos de un niño en concreto y poder visualizar todos los padrinos que ha tenido asociados al mismo.



### EMPLEADO

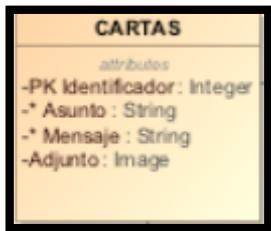
La entidad empleado se elaboró pensando en las peticiones del cliente. Un empleado estará identificado con una PK denominada Usuario. Un empleado se encargará de registrar paquetes y de validar cartas. Además de estas funciones tal y como se comenta al principio, atiende múltiples peticiones de usuario con el fin de conceder o cancelar un apadrinamiento.

Aquí podemos observar los atributos que conformarán la entidad:



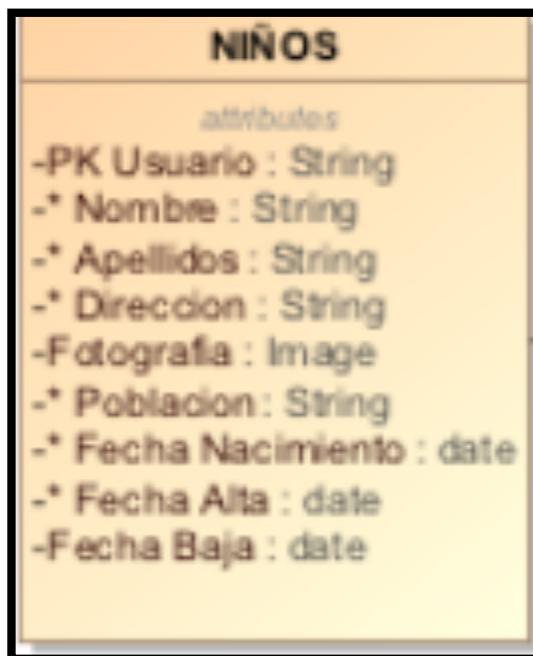
## CARTAS

Esta entidad la construimos con el fin de almacenar un conjunto de cartas identificadas con un identificador(PK), las cuales, van dirigidas a niños y almacenadas en el historial de cartas. No obstante, cada carta es validada por un empleado.



## NIÑOS

Niños es una tabla que almacena información de carácter importante acerca de los niños candidatos a ser apadrinados. Cada niño contendrá un user (Atributo Usuario), el cual será su PK. Dicha entidad establecerá relación con historial padrinos, que será una tabla intermedia entre usuarios y niños, y con cartas (pues son los destinatarios de ellas).



## HISTORIAL PAQUETES

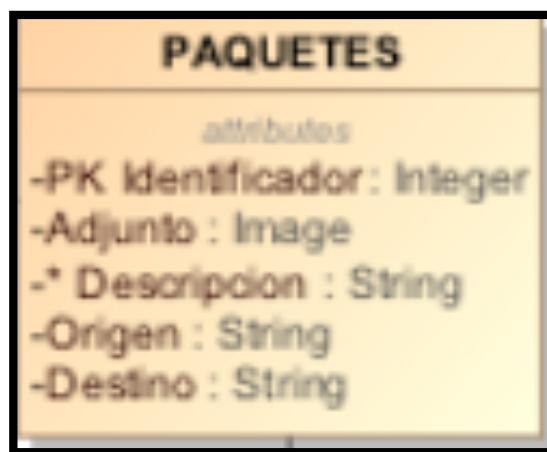
**Historial paquetes** es una tabla que se realizó con el fin de almacenar una serie de historiales sobre los paquetes. Adjuntamos sus atributos:



Un empleado gracias a esta entidad puede visualizar un registro con todos los historiales almacenados, permitiendo la gestión de los mismos. Permite por ejemplo que un paquete se registre en dicho historial y al ser recibido, su estado se modifique a entregado, entre otras muchas funciones recogidas en los requisitos.

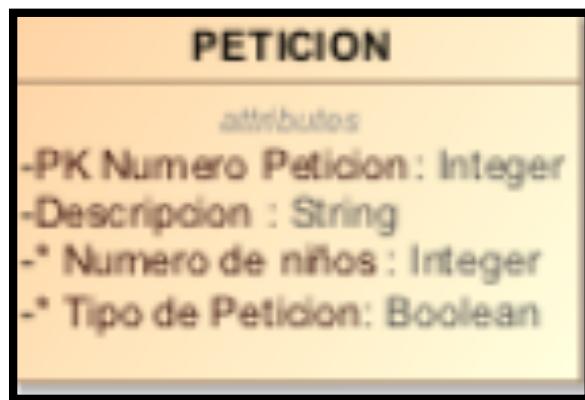
## PAQUETES

**Paquetes** es una tabla, que mediante un identificador (PK) almacena información valiosa sobre los paquetes. Estos paquetes son registrados mediante empleados y conforman un historial de paquetes.



## PETICIÓN

Presentamos ahora la tabla petición, la cual identificaremos con la PK número petición. Esta tabla se implementó con el fin de atender a las peticiones que realizan los usuarios.



Un usuario al solicitar un apadrinamiento o cancelación, rellena una petición que será atendida posteriormente por un empleado, encargado de gestionar la misma y realizar los trámites correspondientes.

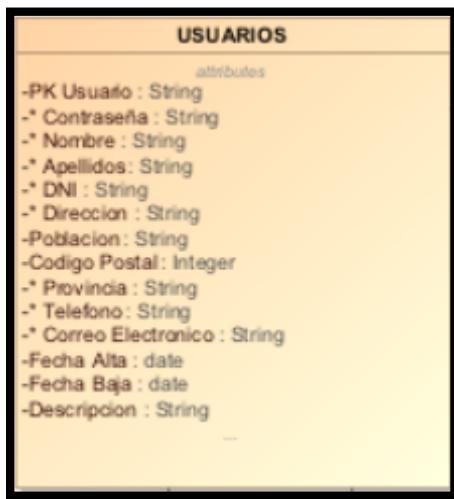
## HISTORIAL CARTAS

Historial cartas es una tabla que se desarrolló atendiendo a que un usuario puede poseer uno o varios historiales de cartas, los cuales, conforman varias cartas. Identificaremos esta entidad con el atributo numero carta y contendrá un atributo booleano Estado para conocer su estado.



## USUARIOS

La presente entidad se realizó con el fin de almacenar datos sobre el usuario. Identificaremos a un usuario de forma inequívoca mediante la PK Usuario. Podemos apreciar en la siguiente ilustración los demás atributos de la tabla.



Gracias a estas entidades, podríamos cumplir los requisitos propuestos en el documento.

Para confirmar la correcta definición establecida en base a los archivos JPA, hemos realizado las siguientes tareas:

```
compile:  
Created dir: /Users/ivan/Documents/GitHub/SII-ACOES-HEFESTO_SOFTWARE/HefestoSoftware/dist  
Copying 1 file to /Users/ivan/Documents/GitHub/SII-ACOES-HEFESTO_SOFTWARE/HefestoSoftware/build  
Copy libraries to /Users/ivan/Documents/GitHub/SII-ACOES-HEFESTO_SOFTWARE/HefestoSoftware/dist/lib.  
Building jar: /Users/ivan/Documents/GitHub/SII-ACOES-HEFESTO_SOFTWARE/HefestoSoftware/dist/HefestoSoftware.jar  
To run this application from the command line without Ant, try:  
java -jar "/Users/ivan/Documents/GitHub/SII-ACOES-HEFESTO_SOFTWARE/HefestoSoftware/dist/HefestoSoftware.jar"  
jar:  
BUILD SUCCESSFUL (total time: 8 seconds)
```

**Vemos que compila correctamente todos los archivos JPA**

A continuación, ejecutamos la clase principal HefestoSoftware:

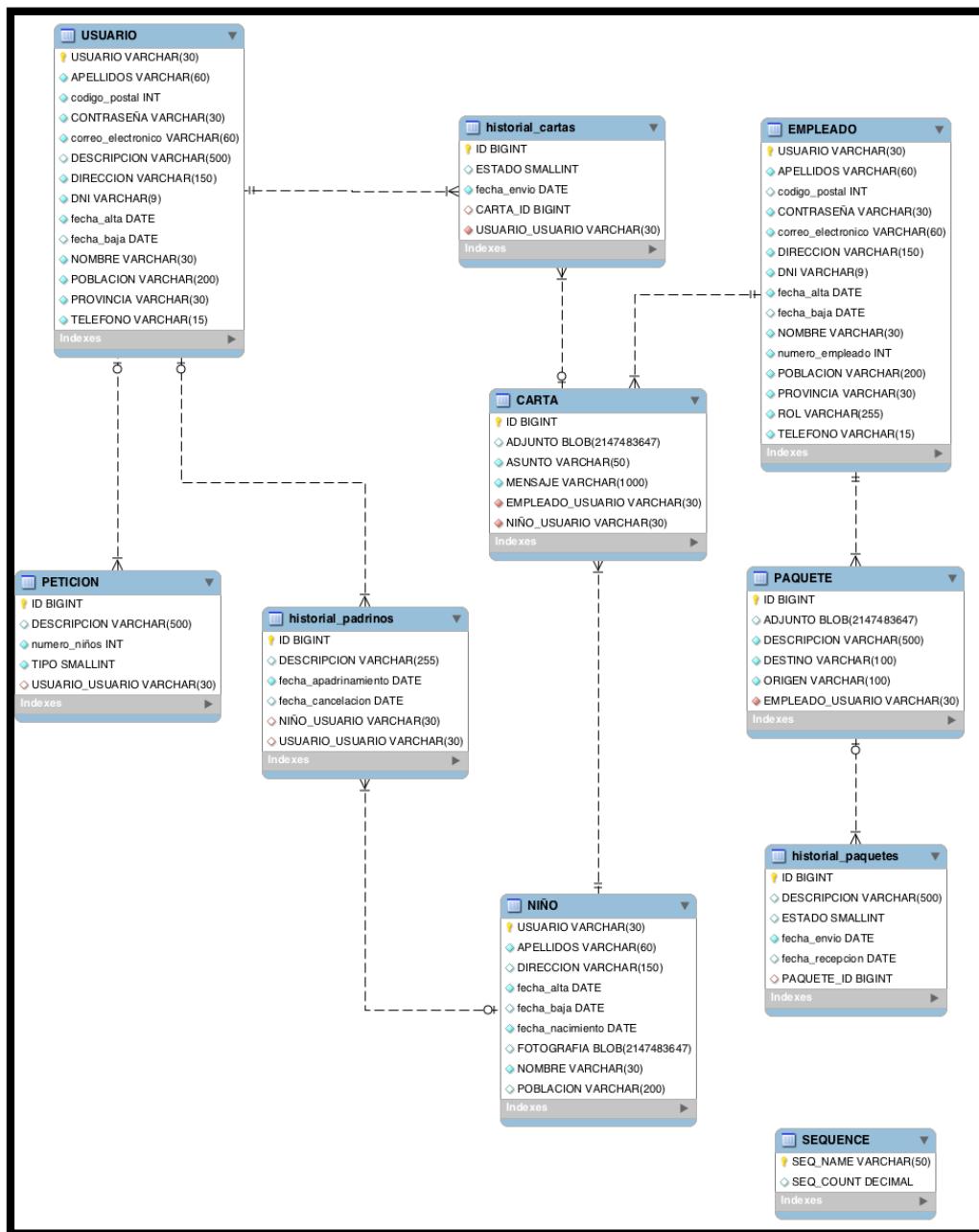
```
compile:  
run:  
[EL Info]: 2019-04-04 13:57:54.798--ServerSession(296347592)--EclipseLink, version: Eclipse Persistence Services - 2.5.2.v20140319-9ad6abd  
[EL Info]: connection: 2019-04-04 13:57:55.646--ServerSession(296347592)--file:/Users/ivan/Documents/GitHub/SII-ACOES-HEFESTO_SOFTWARE/HefestoS  
[EL Info]: connection: 2019-04-04 13:57:57.491--ServerSession(296347592)--file:/Users/ivan/Documents/GitHub/SII-ACOES-HEFESTO_SOFTWARE/HefestoS  
BUILD SUCCESSFUL (total time: 5 seconds)
```

Una vez que dicho proceso se ha creado, nos genera un documento en formato .sql encargado de establecer el esquema definido. Se adjunta a continuación una imagen del mismo.

```
create.sql x
1 CREATE TABLE USUARIO (USUARIO VARCHAR(30) NOT NULL, APELLIDOS VARCHAR(60) NOT NULL, co
2 CREATE TABLE PETICION (ID BIGINT NOT NULL, DESCRIPCION VARCHAR(500), numero_niños INT
3 CREATE TABLE EMPLEADO (USUARIO VARCHAR(30) NOT NULL, APELLIDOS VARCHAR(60) NOT NULL, c
4 CREATE TABLE CARTA (ID BIGINT NOT NULL, ADJUNTO BLOB(2147483647), ASUNTO VARCHAR(50) N
5 CREATE TABLE historial_cartas (ID BIGINT NOT NULL, ESTADO SMALLINT DEFAULT 0, fecha_en
6 CREATE TABLE historial_padrinos (ID BIGINT NOT NULL, DESCRIPCION VARCHAR(255), fecha_a
7 CREATE TABLE NIÑO (USUARIO VARCHAR(30) NOT NULL, APELLIDOS VARCHAR(60) NOT NULL, DIREC
8 CREATE TABLE PAQUETE (ID BIGINT NOT NULL, ADJUNTO BLOB(2147483647), DESCRIPCION VARCHA
9 CREATE TABLE historial_paquetes (ID BIGINT NOT NULL, DESCRIPCION VARCHAR(500), ESTADO
10 ALTER TABLE PETICION ADD CONSTRAINT PTIONSARIOUSUARIO FOREIGN KEY (USUARIO_USUARIO) R
11 ALTER TABLE CARTA ADD CONSTRAINT CARTA_NIÑO_USUARIO FOREIGN KEY (NIÑO_USUARIO) REFEREN
12 ALTER TABLE CARTA ADD CONSTRAINT CARTA_HISTORIAL_ID FOREIGN KEY (HISTORIAL_ID) REFEREN
13 ALTER TABLE CARTA ADD CONSTRAINT CRTAMPLEADOUSUARIO FOREIGN KEY (EMPLEADO_USUARIO) REF
14 ALTER TABLE historial_cartas ADD CONSTRAINT hstrlcartasSRSARIO FOREIGN KEY (USUARIO_USUAR
15 ALTER TABLE historial_padrinos ADD CONSTRAINT hstrlpdrinosSRSRIO FOREIGN KEY (USUARIO_USUAR
16 ALTER TABLE historial_padrinos ADD CONSTRAINT hstrlpadrinosNSRIO FOREIGN KEY (NIÑO_USUAR
17 ALTER TABLE PAQUETE ADD CONSTRAINT PQTEMPLEADOUSUARIO FOREIGN KEY (EMPLEADO_USUARIO) R
18 ALTER TABLE historial_paquetes ADD CONSTRAINT hstrlpaquetesPQTID FOREIGN KEY (PAQUETE_
19 CREATE TABLE SEQUENCE (SEQ_NAME VARCHAR(50) NOT NULL, SEQ_COUNT DECIMAL(15), PRIMARY K
20 INSERT INTO SEQUENCE(SEQ_NAME, SEQ_COUNT) values ('SEQ_GEN', 0);
21
```

Utilizando la herramienta MySQLWorkbench, hemos realizado ingeniería inversa con el fin de poder visualizar el modelo que define este archivo.

Una vez realizado este proceso, podemos confirmar que se ajusta a lo establecido desde un principio. A continuación, se adjunta una imagen, la cual también se ha subido al repositorio de github.



Acto seguido, hemos creado una base de datos local, importando dicho archivo para definir la estructura, visualizar los esquemas generados y realizar algunas operaciones básicas, determinando si el comportamiento definido es el correcto.

select * from historial_padrinos;				
<input type="button" value="Salida de Script"/> <input type="button" value="Resultado de la Consulta"/>				
<input type="button" value="SQL"/>	Todas las Filas Recuperadas: 0 en 0,172 segundos			
ID	DESCRIPCION	FECHA_APADRINAMIENTO	FECHA_CANCELACION	NIÑO_USUARIO
				USUARIO_USUARIO

select * from historial_paquetes;				
<input type="button" value="Salida de Script"/> <input type="button" value="Resultado de la Consulta"/>				
<input type="button" value="SQL"/>	Todas las Filas Recuperadas: 0 en 0,047 segundos			
ID	DESCRIPCION	ESTADO	FECHA_ENVIO	FECHA_RECEPCION
				PAQUETE_ID

select * from historial_cartas;				
<input type="button" value="Salida de Script"/> <input type="button" value="Resultado de la Consulta"/>				
<input type="button" value="SQL"/>	Todas las Filas Recuperadas: 0 en 0 segundos			
ID	ESTADO	FECHA_ENVIO	CARTA_ID	USUARIO...

Tal y como podemos observar, se ha generado un esquema correcto en base a los requisitos definidos, en las anteriores imágenes podemos por ejemplo ver el esquema de los historiales.

Y a continuación, se incluye el esquema de las tablas generadas en SQL:

