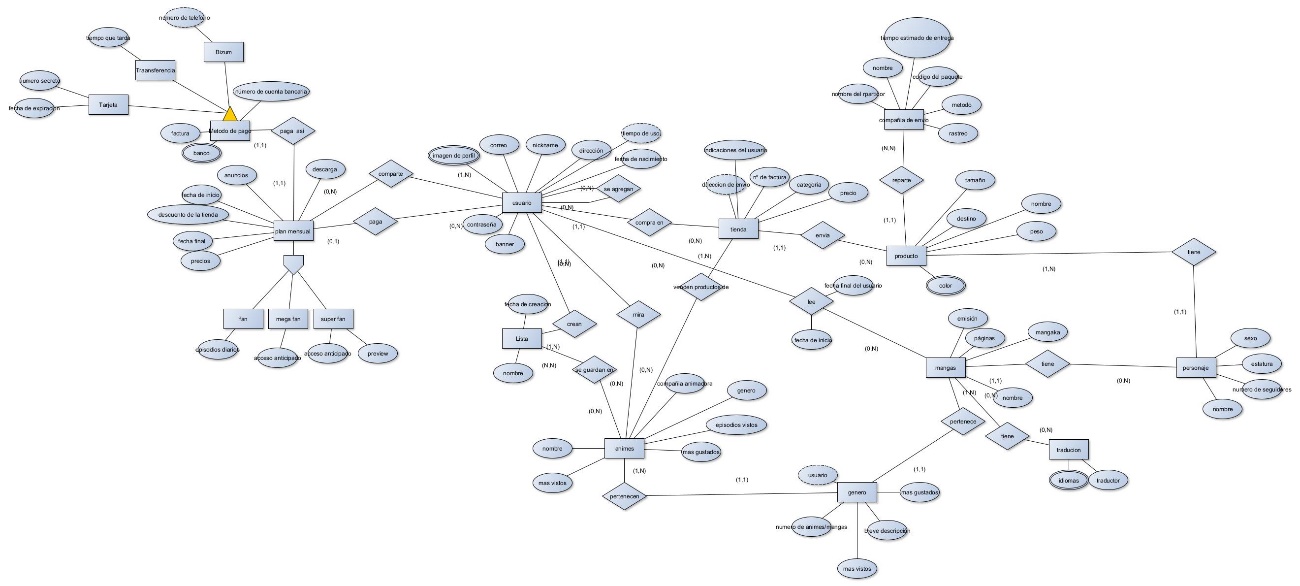
**CRUNCHYROLL**

La empresa crunchyroll quiere actualizar su actual base de datos, está buscando una nueva forma de almacenar los datos para que se más accesible y fácil de aplicar:

* En primer lugar, queremos que el usuario este registrado, pidiéndole el correo y la contraseña de este asi como un nickname, fecha de nacimiento, imagen de perfil, banner, dirección.
* Tenemos que tener en cuenta que la imagen puede ser la misma, pero personalizándole los colores.
* Después se querrá registrar que plan mensual ha elegido el usuario, contando asi con fan, mega fan y super fan, el plan fan tiene un límite diario, el mega fan acceso anticipado y el super fan preview de episodios, guardaremos los precios, y si tiene anuncios ,si dispone de descarga de episodios, y el descuento de la tienda, de esta sección querremos guardar la fecha de inicio de la suscripción y si la tiene del final de esta, asi como si vuelve a activarla, tenemos que tener en cuenta de que el pago puede ser realizado por varios usuarios.
* De los animes que el usuario empieza guardaremos los nombre, episodios vistos, asi como el género de este y la compañía encargada de animarlo para su futura retribución.
* La página cuenta con una tienda donde el usuario obtiene rebajas en su compra por ser premium, de esta guardaremos todas las compras realizadas y de los animes que están relacionados, asi como la categoría del producto, precio, compañía de envió, nº de factura y las posibles indicaciones del usuario.
* Del producto se guardará, la compañía de envió, tamaño, nombre, destino, peso y las unidades en las que esta.
* ****Tenemos que tener en cuenta que los usuarios se pueden hacer amigos de otros usuarios.
* Guardaremos el tiempo de uso de la plataforma asi como los animes más vistos y los más gustados, como lo haremos igual con los géneros de estos.
* Deberemos especificar que usuarios comparten planes, y las listas de cada usuario, de las listas guardaremos los animes, las listas solo pueden ser creadas por un usuario.
* De las listas se guardarán los animes y que usuarios acceden a ellas asi como el nombre y la fecha de creación.
* Guardaremos los mangas leídos por los usuarios, de estos el nombre, genero, mangaka, fecha de inicio, si está en emisión y si no cuando acabó, y cuando lo acabo el usuario y por ultimo las páginas de este y si esta traducido y los idiomas a lo que esta.
* Guardaremos el género de los animes donde deberemos especificar, la cantidad de anime pertenecientes y una breve descripción.
* ****Del producto tenemos que tener en cuenta que pueden ser de los diferentes colores.
* De la tienda se tendrá que tener la dirección de envió del paquete para mandársela a la compañía de repartos.
* De los géneros, tendremos que tener un registro de cada usuario, para saber culés el género que más ve.
* De los planes de usuario se tendrá en cuenta de que los usuarios podrán elegir el método de pegao elegido de este guardaremos el número de factura y el banco y si es tarjeta saber que banco es el número secreto y la fecha de expiración, teniendo 2 formas más, transferencia bancaria (almacenaremos el tiempo que tardó en llegar el dinero) y bizum.
* Guardaremos si los mangas tienen traduciendo esta guardaremos los idiomas en los que están traducidos y quien los ha traducido.
* De los mangas sacaremos los personajes mas gustados y guardaremos su nombre, sexo, estatura y número de seguidores, asi como deberemos saber que código de producto tiene, pudiendo tener solo uno.
* ****Guardaremos que compañía de envió entrega cada paquete, de esta guardaremos el nombre, el tiempo estimado de entrega, el código del paquete, y el método de envió el nombre del repartidor y el código de rastreo del paquete.
* Cada usuario tendrá un estado de ánimo que podrá ser feliz, triste, enfadado, aburrido.
* ****De la compañía de reparto guardaremos las situaciones tráfico siendo las posibilidades libre, concurrido y atasco.

**Diagrama**

**Modelo Relacional:**

plan mensual(#id,anuncios,descargas,fecha\_final,final\_inicio,precios,descuento\_en\_tienda)

fan(#id\_fan,-id\_plan\_mensual,episodios\_diarios)

mega\_fan(#id\_mega\_fan,-id\_plan\_mensual,acceso\_anticipado)

super\_fan(#id\_super\_fan,-id\_plan\_mensual,acceso\_anticipado,preview)

usuario(#id\_usuario,-id\_plan\_mensual,nickname,correo,contraseña,estado\_de\_animo,numero de cuenta bancaria,imagen de perfil,tiempo de uso,fecha de naciemneto,banner,direccion)

plan\_menusal\_usuario(#(-id\_plan\_menusal,-id\_usuario))

metodo de pago(#id,factura,numeor\_de\_cuenta\_bancaria)

banco(#id,banco,-id\_metodo\_de\_pago)

tarjeta(#id,-id\_metodo\_de\_pago,numero\_asociado,fecha\_de\_expiracion)

transferencia(#id,-id\_metodo \_de\_pago,tiempo\_que\_tarda)

bizum(#id,-id\_metodo\_de\_pago,numero\_de\_telefono)

usuario\_usuario(#(-id\_usuario\_agregado,-id\_usuario\_agrega))

imagen\_de\_perfil(#id,-id\_usuario,imagen\_de\_perfil)

lista(#id,-id\_usuario,fecha\_de\_creacion,nombre)

animes(#id-id\_genero,compañia\_animadora,episodios\_vistos,mas\_gustados,mas\_vistos,nombre)

lista\_anime(#(-id\_lista,-id\_anime)

tienda(#id,-id\_usuario,direccion\_de\_correo,nº\_de\_factura,categoria,precio,indicaciones\_del\_usuario)

genero(#id,usuario,numero\_de\_animes\_y/o\_mangas\_vistos,mas\_vistos,breve\_descripcion,mas\_gustados)

animes\_tienda(#(-id\_animes,-id\_tienda))

usuario\_anime(#(-id\_usuario,-id\_anime))

mangas(#id,-id\_genero,emison,nombre,mangaka,paginas)

usuario\_mangas(#(-id\_mangas,-id\_usuarios)fecha\_final\_del\_usuario,fecha\_de\_inicio)

traducion(#id,subtitulos)

idiomas(#id,-id\_traduccion,idioma)

traduccion\_manga(#(-id\_traduccion,-id\_manga))

personaje(#id,-id\_manga,nombre,numero\_de\_seguidores,estatura,sexo)

compañia\_de\_envio(#id,rastreo,metodo,estado\_de\_trafico,codigo\_del\_paquete,tiempo\_estimado\_de\_entrega,nombre,nombre\_del\_reoartidor)

producto(#id,-id\_tienda,-id\_compañia\_de\_envio,-id\_personaje,nombre,tamaño,destino,peso)

color(#id,-id\_producto,color)

**SQL:**

drop database if exists crunchy;

create database if not exists crunchy;

use crunchy;

CREATE USER IF NOT EXISTS 'administrador'@'localhost' IDENTIFIED BY 'administrador';

GRANT ALL PRIVILEGES ON crunchy.\* TO 'administrador'@'localhost' WITH GRANT OPTION;

create table plan\_mensual(

id int unsigned primary key auto\_increment,

anuncion tinyint(1),

descargas varchar(999),

fecha\_final time,

final\_inicio time,

precios tinyint unsigned,

descuento\_en\_tienda tinyint

);

create table fan(

id int unsigned primary key auto\_increment,

id\_plan\_mensual int unsigned,

episodios\_diarios tinyint

);

create table mega\_fan(

id int unsigned primary key auto\_increment,

id\_plan\_mensual int unsigned,

acceso\_anticipado tinyint (1) unsigned

);

create table super\_fan(

id int unsigned primary key auto\_increment,

id\_plan\_mensual int unsigned,

acceso\_anticipado tinyint (1) unsigned,

preview tinyint (1) unsigned

);

create table usuario(

id int unsigned primary key auto\_increment,

id\_plan\_mensual int unsigned,

nickname varchar(20),

correo varchar (20),

contraseña varchar(20),

estado\_animo enum ('feliz', 'triste', 'enfadado','aburrido'),

numero\_de\_cuenta\_bancaria int unsigned not null,

imagen\_de\_perfil varchar (30),

tiempo\_de\_uso float unsigned,

banner varchar (30),

direccion varchar (30)

);

create table plan\_menusal\_usuario(

id\_plan\_mensual int unsigned,

id\_usuario int unsigned,

foreign key ( id\_plan\_mensual) references plan\_mensual(id),

foreign key ( id\_usuario) references usuario(id),

primary key(id\_plan\_mensual, id\_usuario)

);

create table metodo\_de\_pago(

id int unsigned primary key auto\_increment,

factura varchar(30),

numero\_de\_cuenta\_bancaria int unsigned

);

create table banco(

id int unsigned primary key auto\_increment,

id\_metodo\_de\_pago int unsigned,

banco varchar(20)

);

create table tarjeta(

id int unsigned primary key auto\_increment,

id\_metodo\_de\_pago int unsigned,

numero\_asociado int unsigned,

fecha\_de\_expiracion time

);

create table transferencia(

id int unsigned primary key auto\_increment,

id\_metodo\_de\_pago int unsigned,

tiempo\_que\_tarda varchar(20)

);

create table bizum(

id int unsigned primary key auto\_increment,

id\_metodo\_de\_pago int unsigned,

numero\_de\_telefono int unsigned

);

create table usuario\_usuario(

id\_usuario\_agregado int unsigned,

id\_usuario\_agrega int unsigned,

foreign key(id\_usuario\_agregado) references usuario(id),

foreign key(id\_usuario\_agrega) references usuario(id),

primary key(id\_usuario\_agregado,id\_usuario\_agrega)

);

create table imagen\_de\_perfil(

id int unsigned primary key auto\_increment,

id\_usuario int unsigned,

imagen\_de\_perfil varchar(20)

);

create table lista(

id int unsigned primary key auto\_increment,

id\_usuario int unsigned,

fecha\_de\_creacion datetime,

nombre varchar(20)

);

create table genero(

id int unsigned primary key auto\_increment,

id\_usuario int unsigned,

animes\_mangas\_vistos int unsigned,

mas\_vistos varchar(40),

breve\_descripcion varchar(50),

mas\_gustados varchar(40)

);

create table anime (

id int unsigned primary key auto\_increment,

id\_genero int unsigned,

compañia\_animadora varchar(40),

episodios\_vistos varchar(40),

mas\_gustados varchar(40),

mas\_vistos varchar(40),

nombre varchar(20)

);

create table lista\_anime(

id int unsigned primary key auto\_increment,

id\_lista int unsigned,

id\_anime int unsigned

);

create table tienda(

id int unsigned primary key auto\_increment,

id\_usuario int unsigned,

direccion\_de\_correo varchar(30),

nº\_de\_factura int unsigned not null,

categoria varchar(30),

precio double unsigned,

indicaciones\_del\_usuario varchar(80)

);

create table animes\_tienda(

id\_anime int unsigned,

id\_tienda int unsigned,

foreign key (id\_anime) references anime(id),

foreign key (id\_tienda) references tienda(id),

primary key(id\_anime,id\_tienda)

);

create table usuario\_anime(

id\_usuario int unsigned,

id\_anime int unsigned,

foreign key (id\_usuario) references usuario(id),

foreign key (id\_anime) references anime(id),

primary key(id\_usuario,id\_anime)

);

create table mangas (

id int unsigned primary key auto\_increment,

id\_genero int unsigned,

emision tinyint(1) unsigned,

nombre varchar(20),

mangaka varchar(20),

paginas int unsigned

);

create table usuario\_mangas(

id\_usuario int unsigned,

id\_mangas int unsigned,

fecha\_final\_del\_usuario date,

fecha\_de\_inicio date

);

create table traducion(

id int unsigned primary key auto\_increment,

subtitulos tinyint(1) unsigned

);

create table idiomas(

id int unsigned primary key auto\_increment,

idioma varchar(20)

);

create table traduccion\_manga(

id\_manga int unsigned,

id\_traduccion int unsigned

);

create table personaje(

id int unsigned primary key auto\_increment,

id\_manga int unsigned,

nombre varchar(20),

numero\_de\_seguidores int unsigned,

estatura float unsigned,

sexo varchar(10)

);

create table compañia\_de\_envio(

id int unsigned primary key auto\_increment,

rastreo int unsigned,

metodo varchar(20),

codigo\_del\_paquete int unsigned,

estado\_del\_trafico enum('libre', 'concurrido','atasco'),

tiempo\_estimado\_de\_entrega varchar(20),

nombre varchar(20),

nombre\_del\_repartidor varchar(20)

);

create table producto(

id int unsigned primary key auto\_increment,

id\_tienda int unsigned,

id\_compañia\_de\_envio int unsigned,

id\_personaje int unsigned,

nombre varchar(20),

tamaño float unsigned,

destino varchar(20),

peso float unsigned

);

create table color(

id int unsigned primary key auto\_increment,

id\_producto int unsigned,

color varchar(20)

);