

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:

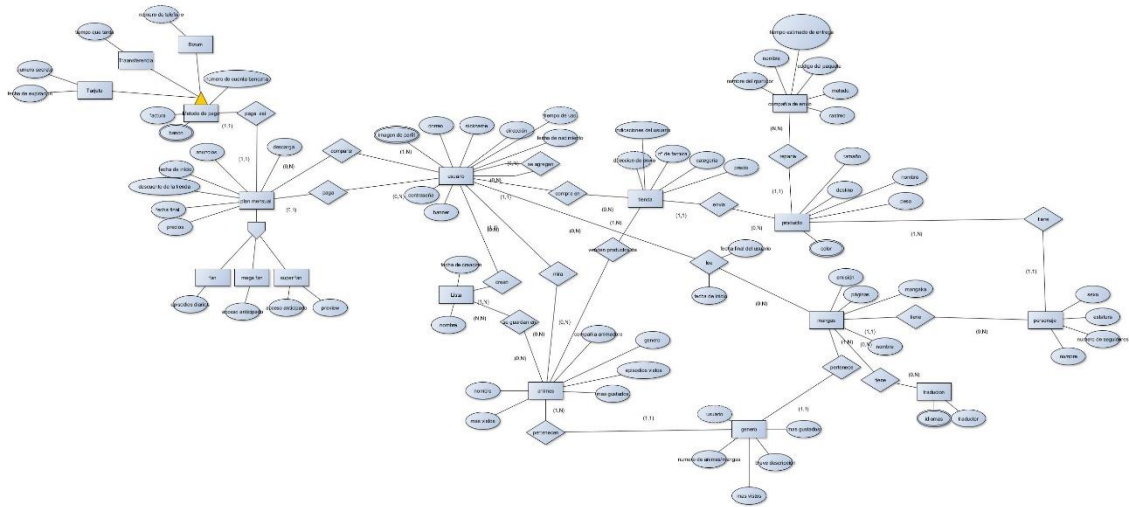
- En primer lugar, queremos que el usuario este registrado, pidiéndole el correo y la contraseña de este asi como un nickname, fecha de nacimiento, imagen de perfil, banner, dirección.
- Tenemos que tener en cuenta que la imagen puede ser la misma, pero personalizándole los colores.
- Después se querrá 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 así como el nombre y la fecha de creación.
- Guardaremos los mangas leídos por los usuarios, de estos el nombre, género, 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 envío del paquete para mandársela a la compañía de repartos.
- De los géneros, tendremos que tener un registro de cada usuario, para saber cuáles el género que más ve.
- De los planes de usuario se tendrá en cuenta de que los usuarios podrán elegir el método de pago 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, así como deberemos saber que código de producto tiene, pudiendo tener solo uno.
- Guardaremos que compañía de envío entrega cada paquete, de esta guardaremos el nombre, el tiempo estimado de entrega, el código del paquete, y el método de envío 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_diaros)

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)

animas(#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_animas_y/o_mangas_vistos,mas_vistos,breve_descripcion,mas_gustados)

animas_tienda(#(-id_animas,-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)

traduccion(#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_mensual_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,
    emission 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 traduccion(
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 compa ia_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)  
);
```

