|  |
| --- |
| Proyecto 2.1 |
|  |

|  |  |
| --- | --- |
| 07/02/2022 | PROYECTO BBDD EN SQL |

|  |
| --- |
| Proyecto BBDD de una aplicación multiplataforma  en SQL Developer.  Realizado por Raúl Benítez Ordóñez. |

# explicación PROYECTO.

Para empezar, vamos a explicar detalladamente en qué consiste la aplicación para la que estamos diseñando la base de datos.

Es necesario saber cómo va a trabajar nuestra aplicación para posteriormente conocer y entender mejor los tipos de datos o restricciones que le estamos dando a nuestra base de datos respecto de esta.

La aplicación que queremos desarrollar es un servicio de música y podcasts, que te da acceso a millones de canciones y podcasts de creadores de todo el mundo.

En la actualidad, disponemos de muchas aplicaciones de este estilo, por ejemplo, Spotify es una de las aplicaciones top en este ámbito, pero nuestra aplicación cuenta con muchas más cosas que esta.



Nuestro principal objetivo es darles a nuestros usuarios **LA MEJOR CALIDAD** en todos los sentidos:

*Los mejores podcasts, las canciones más recientes, la mejor calidad de audio y los mejores precios.*

El funcionamiento de la aplicación sería el siguiente:

Primero de todo necesitaríamos registrarnos para tener una cuenta en la plataforma, y ahí, elegir un tipo de suscripción.

El tipo de suscripción puede ser gratis o de pago. La suscripción de pago ofrece la posibilidad de escuchar canciones y podcasts días o semanas antes de que salgan públicamente en la aplicación, y, elimina completamente la publicidad entre canciones y podcasts.

Después, una vez creada la cuenta, ya tendríamos acceso a las siguientes características:

* Un buscador para explorar por las diferentes canciones y podcasts que podemos escuchar en la aplicación.
* Una biblioteca para almacenar playlists (listas de canciones y podcasts) personalizadas o incluso almacenar canciones y podcasts por separado.
* Un historial donde se mostrarán las últimas canciones y podcasts escuchados por el usuario en los últimos 7 días, por si quiere volver a escuchar algo que no haya guardado previamente.
* Un administrador de usuario donde podrá gestionar sus datos personales, compartir su perfil con otros usuarios e incluso cambiar de suscripción cuando desee.

# creación tablas.

En este apartado vamos a ver cómo creamos las tablas y los diferentes tipos de datos para nuestra base de datos.

Para empezar, vamos a declarar cuales serían las principales tablas que tendríamos que distinguir en nuestra base de datos.

Las tablas serían:

• Tabla “Usuario” donde vamos a registrar a todos los usuarios registrados en la aplicación.

• Tabla “Suscripción” donde vamos a registrar el tipo de suscripción de cada usuario y a su vez, reunir más datos del usuario.

• Tabla “Biblioteca” donde vamos a registrar todas las bibliotecas personales de todos los usuarios.

• Tabla “Historial” donde se vamos a registrar el cajón “historial” de cada usuario.

• Tabla “Playlist” donde vamos a registrar los distintos tipos de playlists de los usuarios con las canciones o podcasts que contienen.

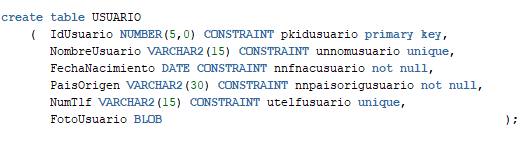
• Tabla “Discos” donde vamos a almacenar todos los álbumes, ep’s o singles que podemos escuchar hasta la fecha en la aplicación.

• Tabla “Canciones” donde vamos a almacenar cada una de las canciones de todos esos discos.

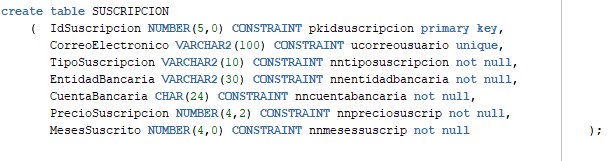
• Tabla “Podcast” donde vamos a almacenar todos los podcasts disponibles para escuchar en la aplicación.

Una vez, distinguidas claramente las tablas con las que vamos a trabajar y su función dentro de la BBDD, ahora necesitamos declarar cada campo y tipo de dato que va a contener cada tabla.

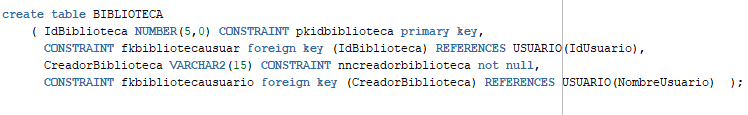
La tabla “Usuario” tendría estos campos:



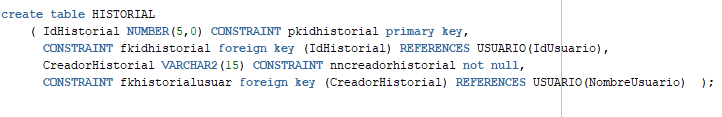
La tabla “Suscripción” tendría estos campos:



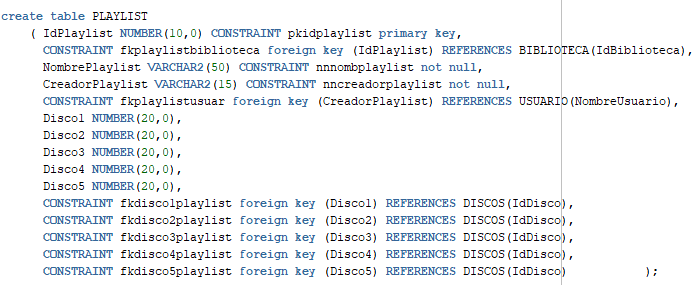
La tabla “Biblioteca” tendría estos campos:



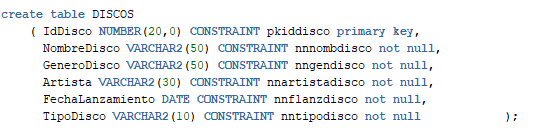
La tabla “Historial” tendría estos campos:



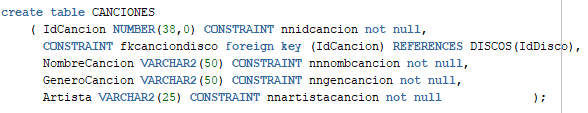
La tabla “Playlist” tendría estos campos:



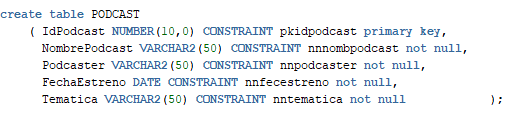
La tabla “Discos” tendría estos campos:



La tabla “Canciones” tendría estos campos:



La tabla “Podcast” tendría estos campos:



# MODIFICACIONES.

Entonces, una vez definidos los campos con sus tipos de datos y restricciones, vamos a ver cómo podemos aplicar algún tipo de modificación para que todos los números de teléfono o los correos electrónicos sigan una estructura.

Vamos a aplicar modificaciones para ciertos campos en las tablas Usuario y Suscripción.

Para estas restricciones vamos a utilizar la sentencia **ALTER TABLE** para modificar la estructura de una tabla que ya existe.

Para añadir una nueva restricción lo vamos a realizar siempre con esta estructura:

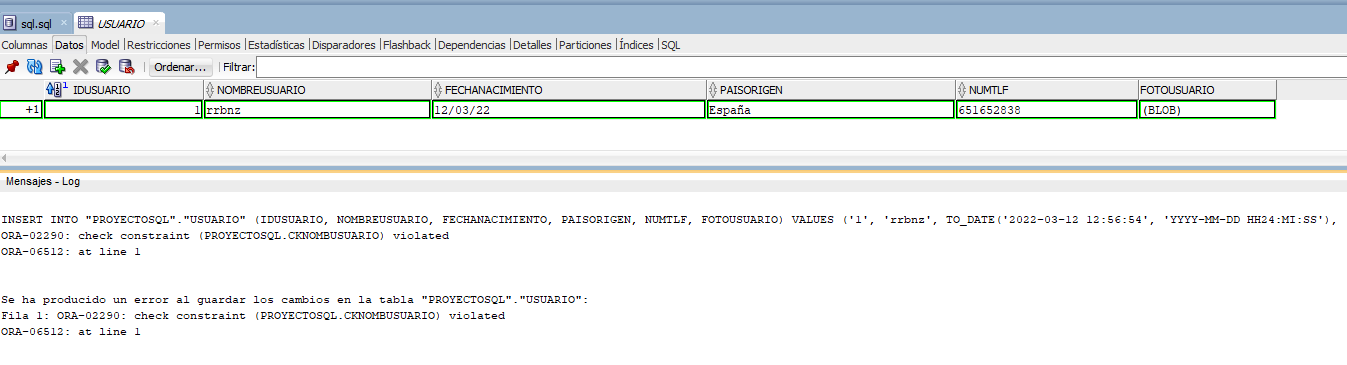


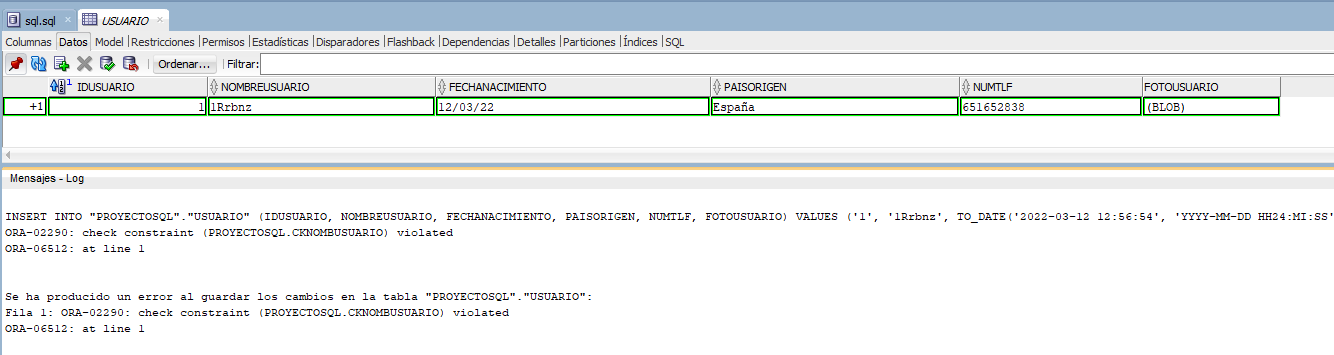
Añadimos la siguiente restricción para el nombre de usuario:



Con esta restricción estamos indicando que el nombre de Usuario tiene que empezar sí o sí con una letra mayúscula, y después especificamos que puede seguir hasta el final de la cadena, con el operador +$, con la clase de caracteres *[:graph:]* que incluye letras mayúsculas, minúsculas, números y hasta signos de puntuación, como guiones, barras bajas, etc.

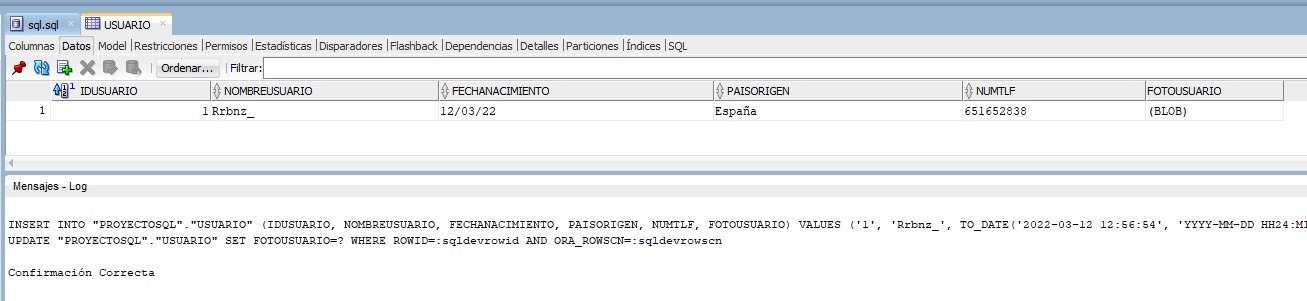
Entonces, aplicamos la restricción en nuestra base de datos y vamos a comprobar que esta funciona correctamente.





Como veis, si al añadir una fila en la BBDD empezamos el usuario con una letra minúscula o con un número, nos salta un mensaje diciendo que estamos violando la restricción que acabamos de añadir a la tabla.

Sin embargo, si empezamos con una letra mayúscula y después añadimos lo que queramos, nos deja perfectamente insertar la fila sin ningún problema.



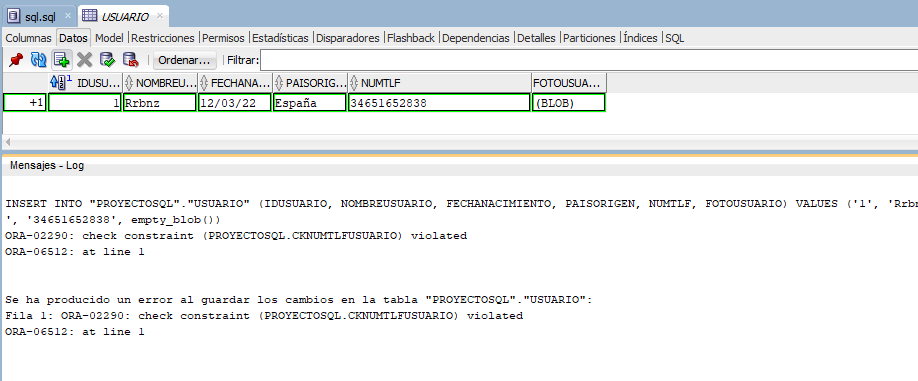
Ahora, para el número de teléfono vamos a utilizar la siguiente restricción:

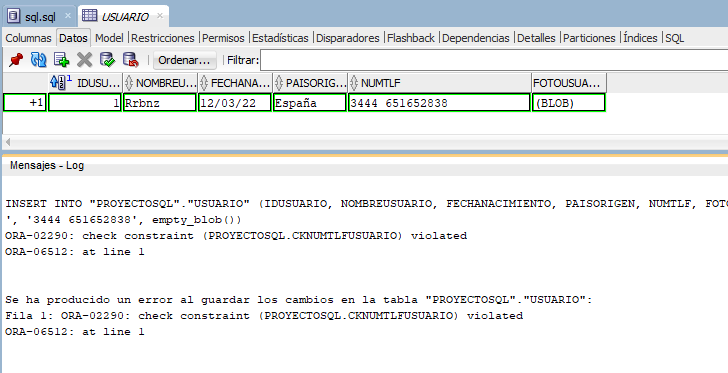


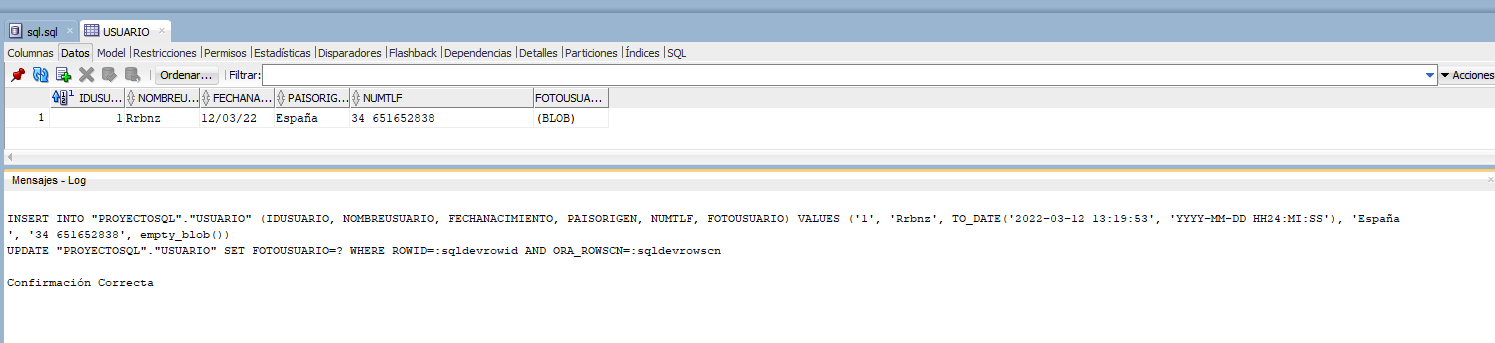
Aquí estamos indicando que el número de tlf va a contener el prefijo del país del que proviene separado con un espacio obligatorio del número del tlf como tal.

Como el prefijo puede variar de uno a tres números, ponemos el rango de {1,3} para poner por ejemplo el prefijo 7 de Rusia o el prefijo 34 de España, el espacio obligatorio que se indica con la clase [:space:] y, por último el número de tlf como tal hasta el final de la cadena.

Entonces, comprobamos la modificación que hemos hecho:







Como veis, si no pongo el espacio o incluso pongo un prefijo de 4 dígitos, me salta la restricción para el número de tlf, pero si lo pongo correctamente, me deja guardar los cambios sin ningún problema.

Por último, vamos a ver dos restricciones que podemos añadir a la tabla Suscripción.

Vamos a modificar los siguientes campos para que el correo electrónico y la cuenta bancaria sigan un patrón.

El patrón que va a seguir el correo electrónico es el siguiente:

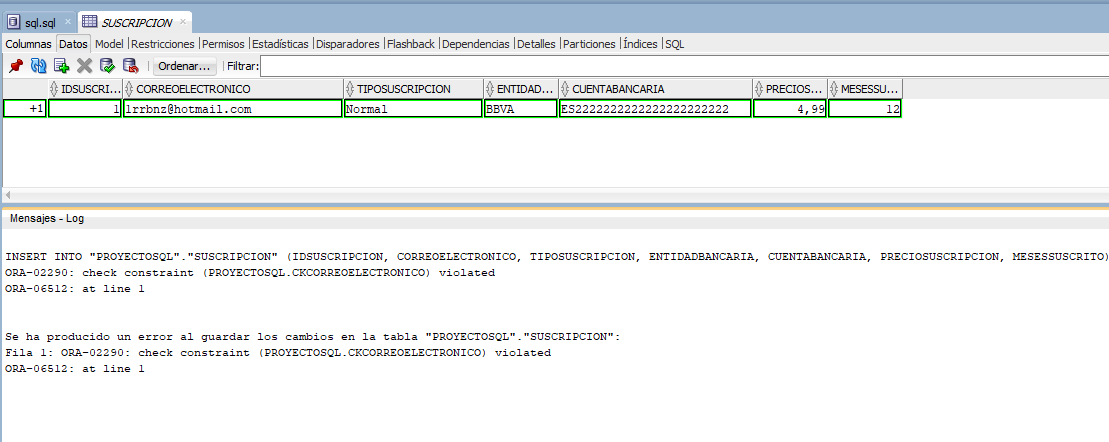


Va a empezar obligatoriamente con un carácter alfabético (ya sea mayúscula o minúscula), después podrá tener desde 4 a 25 caracteres de la clase de caracteres que explicamos antes y después de eso, va a contener obligatoriamente un @.

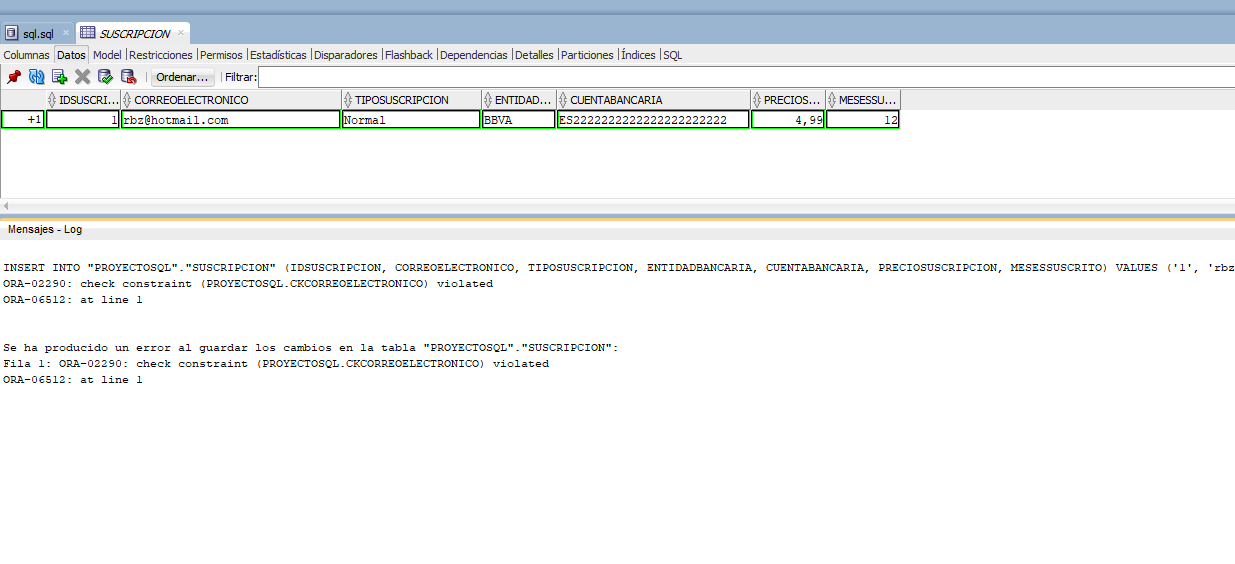
Esto no lo habíamos visto antes, pero podemos indicar específicamente que una cadena empieza por un dígito o carácter específico o incluso que lo contenga a mitad de la cadena.

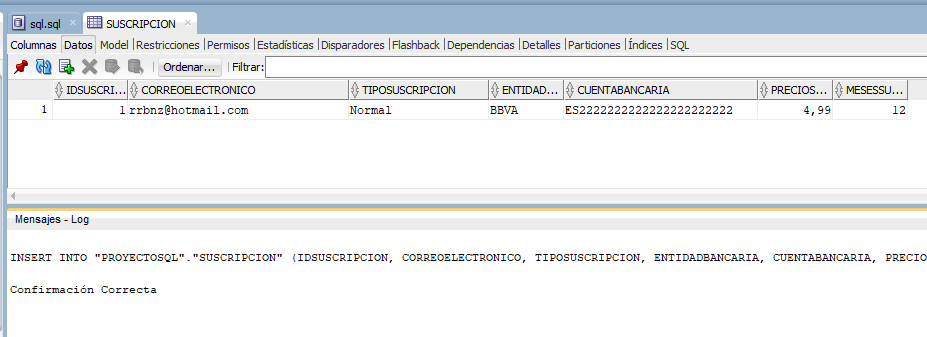
Después del @, va a contener la organización del correo electrónico, como puede ser gmail, yahoo, outlook, etc. y, por último, un punto obligatorio y el tipo de correo electrónico, por ejemplo .com o .es.

En la comprobación, si por ejemplo empezamos por un signo de puntuación o incumplimos el máximo o mínimo de caracteres, nos va a saltar el error, pero si nos ajustamos perfectamente a la cadena, nos deja guardar los datos sin ningún problema.



.





Y para el patrón de la cuenta bancaria, simplemente vamos a decirle que los dos primeros caracteres sean alfabéticos para identificar el código del país y los siguientes 22 dígitos sean únicamente números como en todas las cuentas bancarias.

Entonces, lo hacemos con la siguiente sentencia:



Y, para la comprobación, pues como siempre, pues intentamos introducir datos que no se ajusten a la cadena:

