Manual De Configuración Base de Datos

Nodens

Versión: 0.1.1

*En este documento se especifica la información asociada al proceso de configuración de la o las bases de datos usadas en el proyecto formativo.*

***NOTA:*** *Este y el resto de textos incluidos en cursiva de color azul se vinculan con el fin de proporcionar una guía para el diligenciamiento de este documento y debe ser eliminado antes de realizarse la entrega del mismo. Los textos para el diligenciamiento de la plantilla no deben ser entregados en cursiva y tendrán el color negro, todo el documento deberá contar con el mismo tipo de letra, tamaño y los párrafos deberán estar justificados, no se deberá cambiar la estructura del mismo ni alterar el orden anexando o eliminando secciones sin previa autorización, en caso de que existan secciones que no requieran su diligenciamiento, se debe agregar la sigla NA (No Aplica)*

**HISTORIAL DE REVISIÓN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **VERSIÓN** | **ELABORACIÓN** | | **REVISIÓN** | | **APROBACIÓN** | |
| **Fecha** | **Responsable** | **Fecha** | **Responsable** | **Fecha** | **Responsable** |
| 0.1.1 | 12-11-2022 | Moises Pineda | <dd/mm/aa> | <nombre> | <dd/mm/aa> | <nombre> |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**CAMBIOS RESPECTO A LA VERSIÓN ANTERIOR**

|  |  |
| --- | --- |
| **VERSIÓN** | **MODIFICACIÓN RESPECTO VERSIÓN ANTERIOR** |
| <x.y.z> | <Descripción de la modificación realizada> |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Tabla de contenido**

[1. Introducción 4](#_1fob9te)

[2. Alcance 4](#_3znysh7)

[3. Responsables e involucrados 4](#_2et92p0)

[4. Modelo Entidad Relación (MER) 4](#_tyjcwt)

[5. Diccionario de Datos. 4](#_3dy6vkm)

[6. Modelo Relacional. 4](#_1t3h5sf)

[7. Justificación Motor Seleccionado 4](#_4d34og8)

[8. Requisitos de Configuración 4](#_2s8eyo1)

[9. Scripts 4](#_17dp8vu)

[10. Configuración y Ejecución de la Base de Datos 4](#_3rdcrjn)

[11. Otras Consideraciones 5](#_26in1rg)

# 1. Introducción

Este documento describirá la forma en la que se diseñó la base de datos, las razones por las cuales se escogieron las tecnologías en las que se desarrollan, las configuraciones para su ejecución y su script de creación.

Se especificarán todas las bases de datos usadas en el proyecto, ya que la arquitectura es de microservicios explicando todos los ítems correspondientes a una base de datos.

# 2. Alcance

# 2.1. Autenticación e Información de Usuario

Esta base de datos se encargará de almacenar la información básica del registro y la información de autenticación y el rol del usuario.

# 2.2. Manejo de ofertas

Esta base de datos almacenará todas las ofertas que se publiquen y que el organizador quiera mantener.

# 2.3. Manejo de la información de los aplicantes

Esta base de datos tendrá la información relacionada al nivel, skills, experiencia y demás información sobre el músico o agrupación.

# 2.4 Manejo de Post

Esta base de datos tendrá publicaciones de propósito general para brindar información.

# 2.5 Manejo de la información de los organizadores.

Esta base de datos tendrá la información de los organizadores o las personas que hagan ofertas.

# 2.6 Verificación de cuenta y Recuperación de contraseña.

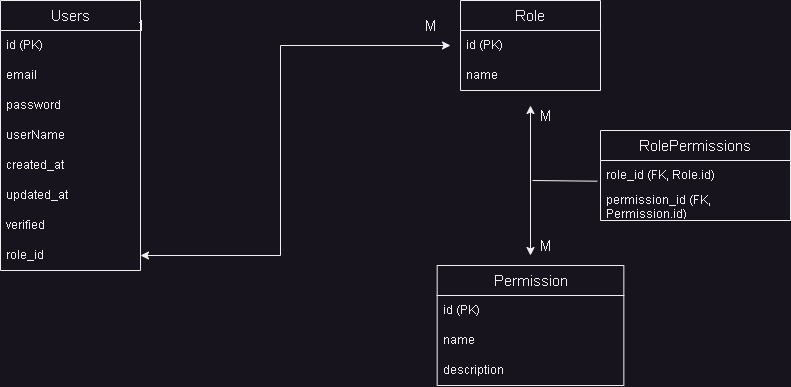
Esta base de datos almacenará tokens y usuarios que expirarán después de un tiempo, los tokens servirán tanto para recuperar la contraseña como para verificar el correo electrónico de un usuario, el token se generará cuando el usuario requiera cambiar la contraseña o verificar el correo electrónico en caso de que el enlace que se envía al registrar expire.

# 3. Responsables e involucrados

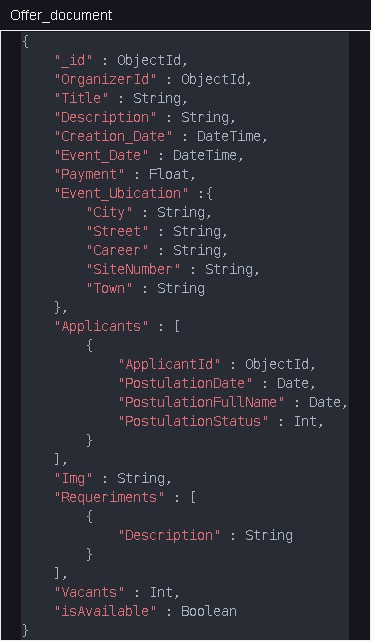
|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo (Responsable/ Involucrado)** | **Rol** |
| Moises Pineda Hernández | Responsable | Backend Dev / Base de Datos |
| Ethiem Alexander Guerrero | Responsable | Frontend Dev / Base de Datos |
| Sebastián García Correa | Responsable | Frontend Dev / Base de Datos |
| Julio Marín Marín | Responsable | SCRUM Master / Base de Datos |
| Brayan Medina Moreno | Responsable | Backend Dev / Base de Datos |

# 4. Modelo Entidad Relación (MER)

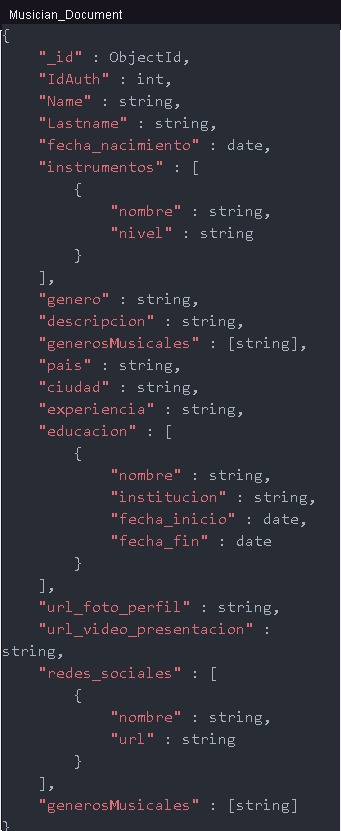
# 4.1. Autenticación e Información de Usuario



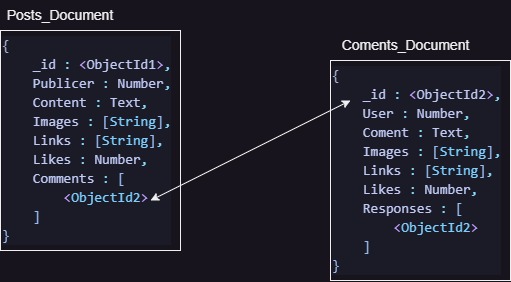
# 4.2. Manejo de ofertas



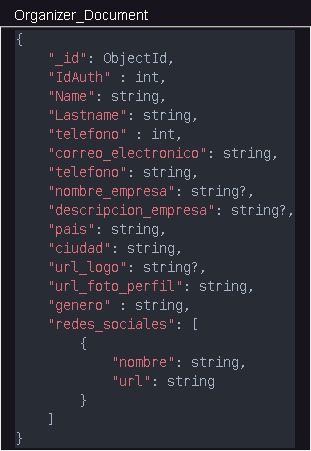
# 4.3. Manejo de la información de los aplicantes



# 4.4 Manejo de post



# 4.5 Manejo de organizadores



# 4.6 Verificación de cuenta y Recuperación de contraseña.

# 

# 5. Diccionario de Datos.

Aquí podemos encontrar las definiciones de los datos y sus propiedades (atributos , dominios, asociaciones y operaciones). Asimismo identificar claramente el propósito, alcance y campo de aplicación ,sirviendo a los usuarios como orientación tipo informativa.

**5.1 Autenticación e información de usuario**

**USERS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | Descripcion | Caracteristicas | Tipo de Dato | Dominio de valores | Indice y relaciones |
| Id | Entero que identifica a un usuario unico | Primary Key  Autoincrement  Not Null | Int | Se define por frecuencias | Primary Key  Autoincrement  Not Null |
| Email | String que identifica al usuario a través de un correo | Campo obligatorio, no acepta valores nulos y tiene que ser unico | Varchar | Correo del usuario | Not null |
| Firts\_Name | String del primer nombre del usuario | Campo obligatorio que no acepta valores nulos | varchar | Nombre del usuario | Not null |
| Second\_name | String del segundo nombre del usuario | Campo opcional que no acepta valores nulos | varchar | Nombre del usuario | Null |
| First\_Lastname | String del primer apellido del usuario | Campo obligatorio que no acepta valores nulos | varchar | Apellido del usuario | Not null |
| Second\_name | String del segundo apellido del usuario | Campo opcional que no acepta valores nulos | varchar | Apellido del usuario | Null |
| Ubication | String de la ubicación o zona de residencia del usuario | Campo obligatorio que no acepta valores nulos | varchar | Ubicación del usuario | Not null |
| Rol | String que define el tipo de rol que tiene el usuario que puede ser organizador o musico | Campo obligatorio que no acepta valores nulos | varchar | Rol del usuario | Not null |
| Birthdate | Date que define la fecha de naciemiento | Campo obligatorio que no acepta valores nulos | Date | Fecha de nacimiento del usuario | Not null |

**Auth\_Users**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | Descripcion | Caracteristicas | Tipo de Dato | Dominio de valores | Indice y relaciones |
| Email | String que identifica al usuario a través de un correo | Foreign key  Campo obligatorio, no acepta valores nulos y tiene que ser unico | Varchar | Correo del usuario | Foreign key  Not null |
| Password | String que permite al usuario identificarse para entrar en la plataforma | Campo obligatorio, no acepta valores nulos y tiene que ser unico | varchar | Contraseña del usuario | Not null |

**5.1 Manejo de Ofertas y organizadores**

**Offer\_document**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | Descripcion | Caracteristicas | Tipo de Dato | Dominio de valores | Indice y relaciones |
| \_id | Number que identifica el post | Number campo que se auto genera | Number | Id del post | Identificador principal not null |
| Title | String titulo del post | String campo obligatorio que no acepta valores nulos | String | Titulo del post | Not null |
| Description | String que describe el post del evento | String campo obligatorio que no acepta valores nulos | String | Descripcion del post | Not null |
| Creation\_Date | DateTime que describe cuando se creo el post | DateTime Que se autogenera | DateTime | Fecha de Creación del post | Not null |
| Event\_Date | DateTime que describe cuando se es la fecha del evento | DateTime campo obligatorio que no acepta valores nulos | DateTime | Fecha del evento | Not null |
| Payment | Number que indica el pago por tocar en el evento | Number campo obligatorio que no acepta valores nulos | Number | Pago por el evento | Not null |
| OrganizerId | Number que indica el usuario con rol de organizador que creo el post | Number campo obligatorio que no acepta valores nulos | Number | Identificador del autor del post | Not null |
| Organization | String que indica la organización a la que pertenece el organizador que hace el post | String campo opcional | String | Organización del autor del post | null |
| Event\_Ubication | Objeto que describe la ubicación donde ser el evento | Object con campos obligatorios y opcionales | Object | Ubicacion del evento | Not null |
| Event\_Ubication.City | String que indica la ciudad donde se realizara el evento | String campo obligatorio que no acepta valores nulos | String | Ciudad del evento | Not null |
| Event\_Ubication.Street | String que indica en que calle se realizara el evento | String campo opcional que no acepta valores | String | Calle del evento | Not null |
| Event\_Ubication.Career | String que indica en que carrera se realizara el evento | String campo opcional que no acepta valores | String | Carrera del evento | Not null |
| Event\_Ubication.SiteNumber | Number que indica en que Numero se realizara el evento | Number campo opcional que no acepta valores | Number | Numero del evento | Not null |
| Event\_Ubication.Town | String que indica en que Barrio se realizara el evento | String campo opcional que no acepta valores | String | Barrio del evento | Not null |
| Applicants | Number que indica la relación con los aplicantes que se han registrado para el evento | Number campo obligatorio que no acepta valores nulos | Number | Identificador de los aplicantes para el post | Not null |
| Img | String que guarda la Url de la imagen representativa del evento | String campo opcional | String | Url de la imagen del post | Null |
| Requirements | Array que guarda los objetos para los requerimientos de cada instrumento | Array con objetos que tienen campos obligatorios | Array | Requerimientos de cada instrumento necesario | Not null |
| Requirements[x].Description | String que describe lo que se necesita | String campo obligatorio que no acepta valores nulos | String | Descripción de lo que se necita | Not null |
| Requirements[x].  ExperienceTime.Years | Number que describe los años de experiencia que debe tener los músicos que se requieren 0 | Number campo obligatorio que no acepta valores nulos | Number | Tiempo que se pide para los aplicantes | Not null |
| Requirements[x].  ExperienceTime.Events\_Played | Number que describe la cantidad de eventos mínimos en los que debe haber participado el musico | Number campo obligatorio que no acepta valores nulos | Number | Eventos que se piden para los aplicantes | Not null |
| Requirements[x].Instrument | String que describe que instrumento se necesita | String campo obligatorio que no acepta valores nulos | String | Intrumento que se requiere | Not null |
| Vacants | Number que describe el numero de vacantes que se piden para el evento | Number campo obligatorio que no acepta valores nulos | Number | Cantidad de vacantes que se piden | Not null |

**Applicants\_Document**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | Descripcion | Caracteristicas | Tipo de Dato | Dominio de valores | Indice y relaciones |
| \_id | Number que indetifica el grupo de aplicantes que están para ese evento | Number campo auto generado | Number | Identificador del grupo de aplicantes | Not null |
| Applicants | Array de objetos que identifican a los postulantes para el evento | Array com campos obligatorios | Array | Información que identifica a cada aplicante | Not Null |
| Applicants.ApplicantId | Number que identifica al potulante para el post | Number Campo obligatorio que no recibe valores nulos | Number | Identificador del aplicante | Not null |
| Applicants.PostulationDate | DateTime que describe la fecha en la que el aplicante se postulo para el evento | DateTime campo obligatorio autogenerado | DateTime | Fecha de postulación | Not null |

**5.3 Manejo de la información de los aplicantes**

**Musician\_Document**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | Descripcion | Caracteristicas | Tipo de Dato | Dominio de valores | Indice y relaciones |
| \_Id | Number identificador del musico | Number campo auto generado | Number | Id del musico | Not null |
| MusicianId | Number que relaciona lainformacion del musico con la infomacion de registro | Number campo obligatorio que no acepta valores nulos | Number | ID de conexión entre musico y datos de registro | Not null |
| YearsExperience | Number que indica los años de experiencia que tiene el musico | Number campo obligatorio que no acepta valores nulos | Number | Años de experiencia | Not null |
| EventsPlayed | Number que indica la cantida de eventos en los que a participado el musico | Number campo obligatorio que no acepta valores nulos | Number | Eventos partipados | Not null |
| Instruments | Number que hace referencia al documento donde se dicen los instrumentos del musico | Number campo que se genera automáticamente | Number | Id de referencia | Not null |
| ProfesionalTitles | Number que hace referencia al documento donde se dicen los Titulos profesionales del musico | Number campo que se genera automáticamente | Number | Id de referencia | Not null |

**Instruments\_Document**

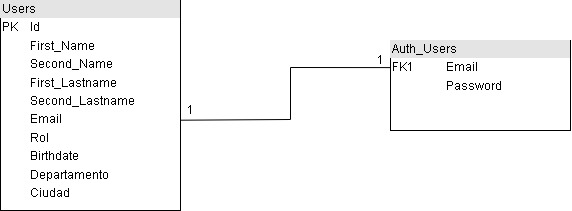
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | Descripcion | Caracteristicas | Tipo de Dato | Dominio de valores | Indice y relaciones |
| \_Id | Number que identifica los instrumentos del musico | Number campo generado automaticamente | Number | Id de los instrumentos | Not null |
| Instruments | Array de objetos donde se describen los instrumentos del musico | Array con campos obligatorios | Array de objetos | Instrumentos del musico | Not null |
| Instruments[x].Name | String que describe el nombre del instrumento | String campo obligatorio que no acepta valores nulos | Srtring | Nombre del instrumento | Not null |
| Instruments[x].InstrExperience | Number que describe los años de experiencia que tiene el musico con ese intrumento | Number campo obligatorio que no acepta valores nulos | Number | Años de experiencia | Not null |
| Instruments[x].Videos | Array de strings con las Urls de los videos para ver como se desempeña el musico con dicho instrumento | Array campo obligatorio que no acepta valores nulos | Array | Videos de comprobación | Not null |

**Titles\_document**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atributo** | Descripcion | Caracteristicas | Tipo de Dato | Dominio de valores | Indice y relaciones |
| \_Id | Number que identíficafica los títulos del musico | Number campo generado automáticamente | Number | Id de los titulos | Not null |
| Titles | Array de objetos con los títulos que tiene el musico | Array de objetos con campos obligatorios | Array de objetos | Titulos del musico | Not null |
| Titles[x].Institution | String que describe la institución donde el musico recibió dicho titulo | String campo obligatorio que no capta valores nulos | String | Nombre de la institución | Not null |
| Titles[x].ProfTitle | String que describe el titulo recibido por el musico | String campo obligatorio que no acepta valores nulos | String | Nombre del titulo | Not null |
| Titles[x].Certificate | String con la Url donde se puede ver el certificado del titulo del musico | String campo obligatorio que no acepta valores nulos | String | Url del certificado | Not null |

# 6. Modelo Relacional.

# 6.1. Autenticación e Información de Usuario



# 7. Justificación Motor Seleccionado

# 7.1. Autenticación e información de Usuario

Se escogió para esta base de datos el motor de SQL Server con bases de datos en el lenguaje de SQL por su consistencia y el alcance que tendrá: la información que almacenará esta base de datos en relación al proyecto no será de alta magnitud en tamaño y cantidad pero si será de suma importancia por lo que se requiere alta consistencia en los datos,

# 7.2. Manejo de ofertas y organizadores

Se eligió para esta base de datos el motor de MongoDB por su rapidez, escalabilidad y flexibilidad puesto que mucha información será guardada allí y los datos que contengan pueden variar según su propósito en el aplicativo así que se requiere rapidez y alta escalabilidad en la base de datos.

# 7.2. Manejo de la información de los aplicantes

Se eligió para esta base de datos el motor de MongoDB por su rapidez, escalabilidad y flexibilidad puesto que mucha información será guardada allí y los datos que contengan pueden variar según su propósito en el aplicativo así que se requiere rapidez y alta escalabilidad en la base de datos.

# 8. Requisitos de Configuración

*Se deben mencionar los requisitos en cuanto a herramientas necesarias o recomendadas para la configuración de la base de datos (Motor de BD, Cliente BD, o Server)*

# 9. Scripts

# 9.1. Autenticación e Información de Usuario

create database AuthServiceDB;

use AuthServiceDB;

create table Role(

id int primary key identity(1,1),

name varchar(50) not null

);

create table Users(

id int primary key identity(1,1),

email varchar(320) unique not null,

password varchar(max) not null,

userName varchar(100) unique not null,

created\_at date not null,

updated\_at date,

Verified bit not null default 0,

role\_id int not null,

constraint fk\_role\_1 foreign key (role\_id) references Role(id)

);

create table Permission(

id int primary key identity(1,1),

name varchar(100) not null,

description varchar(500) not null

);

create table RolePermissions(

role\_id int not null,

permission\_id int not null,

constraint fk\_role\_2 foreign key (role\_id) references Role(id),

constraint fk\_permission foreign key (permission\_id) references Permission(id)

);

--------------------------------------------------------------------------------------------------------

INSERT into Role(name) values ('Musician'),('Organizer')

--------------------------------------------------------------------------------------------------------

go

create procedure SP\_CreateUser

@Email varchar(320),

@userName varchar(100),

@Password varchar(max),

@Role varchar(50)

as

begin transaction TX\_New\_User

BEGIN TRY

if @Role = 'Musician' or @Role = 'Organizer'

begin

INSERT INTO Users(email,password, userName, created\_at, role\_id)

values (@Email, @Password, @userName,GETDATE(), (select id from Role where name = @Role))

COMMIT TRANSACTION TX\_New\_User

SELECT 'Usuario Creado Correctamente' as Message, 0 as Error

end

else select 'Roles invalidos' as Message, 1 as Error

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION TX\_New\_User

SELECT ERROR\_MESSAGE() as Message, 1 as Error

END CATCH

go

---------------------------------

go

create procedure SP\_ReadUser

@Email varchar(320) = NULL,

@username varchar(100) = NULL,

@Id int = NULL

as

begin

SELECT userName, (select name from Role where id = role\_id) as Role, email, Verified from Users where email = @Email or userName = @username or id = @Id

end

go

---------------------------------

go

create procedure SP\_AuthUser

@Email varchar(320)

as

begin

if exists(SELECT id from Users where email = @Email)

SELECT Users.password as 'password', Users.Verified as 'Verified', Users.id as 'id', Users.userName as 'userName', Role.name as 'Role' from Users inner join Role on Users.role\_id = Role.id where email = @Email

else

SELECT null as password, null as Verified, null as Role

end

go

-----------------------

go

create procedure SP\_VerifyUser

@Email varchar(320)

as

begin transaction TX\_VerifyUser

BEGIN TRY

UPDATE Users set Verified = 1, updated\_at = GETDATE() where email = @Email

COMMIT TRANSACTION TX\_VerifyUser

SELECT 'Usuario verificado correctamente' as Message, 0 as Error

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION TX\_VerifyUser

SELECT ERROR\_MESSAGE() as Message, 1 as Error

END CATCH

go

-----------------------

go

create procedure SP\_ChangePassword

@Email varchar(320),

@NewPass varchar(max)

as

begin transaction TX\_ChangePassword

BEGIN TRY

UPDATE Users set password = @NewPass, updated\_at = GETDATE() where email = @Email

COMMIT TRANSACTION TX\_ChangePassword

SELECT 'Contraseña actualizada correctamente' as Message, 0 as Error

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION TX\_ChangePassword

SELECT ERROR\_MESSAGE() as Message, 1 as Eror

END CATCH

go

----------------------

go

create procedure SP\_DeleteUser

@Email varchar(320)

as

begin transaction TX\_DeleteUser

BEGIN TRY

declare @id int

SELECT @id = id from Users where email = @Email

DELETE FROM Users where id = @id

COMMIT TRANSACTION TX\_DeleteUser

SELECT 'Usuario eliminado correctamente' as Message, 0 as Error

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION TX\_DeleteUser

SELECT ERROR\_MESSAGE() as Message, 1 as Eror

END CATCH

go

-------------------

go

create procedure SP\_UpdateUser

@Email varchar(320),

@newEmail varchar(320),

@userName varchar(100)

as

begin transaction TX\_Update\_User

BEGIN TRY

update Users set

userName = ISNULL(@userName, userName),

updated\_at = GETDATE()

where email = @Email

if (@newEmail != @Email) begin

update Users set Verified = 0, email = @newEmail where email = @Email

end

COMMIT TRANSACTION TX\_Update\_User

SELECT 'Usuario actualizado correctamente' as Message, 0 as Error

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION TX\_Update\_User

SELECT ERROR\_MESSAGE() as Message, 1 as Error

END CATCH

go

------------------

# 10. Configuración y Ejecución de la Base de Datos

*En esta sección se describe paso a paso el proceso de configuración de la base de datos, si es necesario especificar el proceso de instalación del motor, el cliente de BD o del server, en caso contrario hacer referencia a manuales externos que deben estar incluidos en los anexos, se debe mostrar el proceso de ejecución del Script y evidenciar la creación de la BD*

# 11. Otras Consideraciones

*En caso de ser necesario se deben indicar aspectos a considerar para la correcta configuración de la BD*