

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Aplicación para la Generación Automatizada**

**de un Diccionario de Datos**

Curso: Patrones de Software

Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

Daleska Nicolle Fernandez Villanueva                     (2021070308)

Andree Sebastián Flores Meléndez                           (2017057494)

Mario Antonio Flores Ramos                                    (2018000597)

**Tacna – Perú**

**2025**

***Aplicación para la Generación Automatizada de un Diccionario de Datos***

**Versión *2.0***

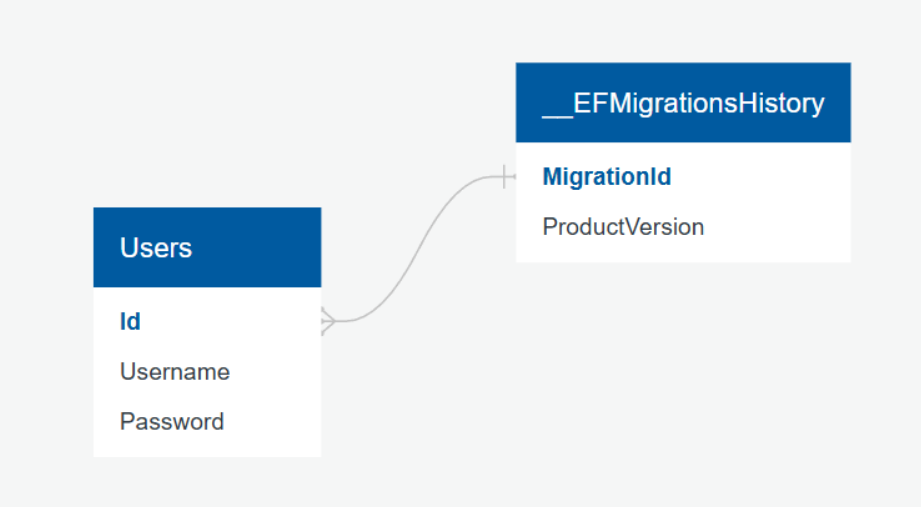
**ÍNDICE GENERAL**

1. [Modelo Entidad / relación 4](#_TOC_250004)
   1. [Diseño lógico 4](#_TOC_250003)
   2. [Diseño Físico 4](#_TOC_250002)
2. [DICCIONARIO DE DATOS 4](#_TOC_250001)
   1. [Tablas 4](#_TOC_250000)
   2. Procedimientos Almacenados 4
   3. Lenguaje de Definición de Datos (DDL) 5
   4. Lenguaje de Manipulación de Datos (DML) 5

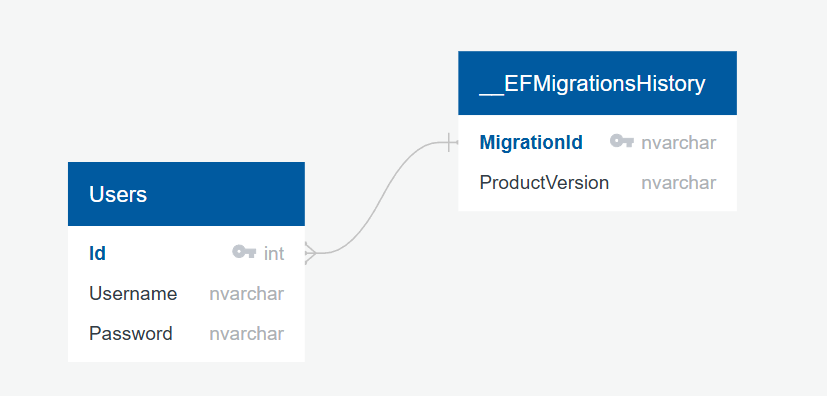
**Diccionario de Datos**

# Modelo Entidad / relación

# Diseño lógico

****

# Diseño Físico

****

# DICCIONARIO DE DATOS

# Tablas

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre de la Tabla:** | | | dbo. EFMigrationsHistory | | | | |
| Descripción de la Tabla: | | | La tabla ' EFMigrationsHistory' almacena el historial de migraciones de Entity Framework, registrando el identificador de cada migración (MigrationId) y la versión del producto (ProductVersion) utilizada. | | | | |
| Objetivo: | | | La tabla ' EFMigrationsHistory' almacena el historial de migraciones aplicadas en una base de datos utilizando Entity Framework, permitiendo el seguimiento de cambios en el esquema. | | | | |
| Relaciones con otras tablas: | | | Sin relaciones detectadas. | | | | |
| **Descripción de los campos** | | |  | | | | |
| **Nro** | **Nombre del Campo** | **Tipo dato longitud** | | **Permite nulos** | **Clave primaria** | **Clave foránea** | **Descripción del campo** |
| 1 | MigrationId | nvarchar(150) | | No | Sí | No | Identifica de manera única cada migración aplicada en Entity Framework. |
| 2 | ProductVersion | nvarchar(32) | | No | No | No | La columna 'ProductVersion' almacena la versión del paquete de Entity Framework utilizado en migraciones. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre de la Tabla:** | | | dbo.Users | | | | |
| Descripción de la Tabla: | | | La tabla 'Users' almacena información de los usuarios del sistema, incluyendo un identificador único (Id), el nombre de usuario (Username) y la contraseña (Password) para la autenticación. | | | | |
| Objetivo: | | | El propósito general de una tabla 'Users' en una base de datos empresarial es almacenar información de identificación y autenticación de los usuarios para gestionar accesos y permisos en el sistema. | | | | |
| Relaciones con otras tablas: | | | Sin relaciones detectadas. | | | | |
| **Descripción de los campos** | | |  | | | | |
| **Nro** | **Nombre del Campo** | **Tipo dato longitud** | | **Permite nulos** | **Clave primaria** | **Clave foránea** | **Descripción del campo** |
| 1 | Id | int | | No | Sí | No | Identificador único para cada usuario, garantizando la integridad y referencia en la base de datos. |
| 2 | Username | nvarchar(-1) | | No | No | No | Identifica de manera única a cada usuario en la base de datos. |
| 3 | Password | nvarchar(-1) | | No | No | No | Almacenar de manera segura las contraseñas de acceso de los usuarios autenticados. |

# Lenguaje de Definición de Datos (DDL)

*-- EFMigrationsHistory –*

CREATE TABLE dbo. EFMigrationsHistory ( MigrationId nvarchar(150) NOT NULL PRIMARY KEY, ProductVersion nvarchar(32) NOT NULL

);

*-- Users –*

CREATE TABLE dbo.Users (

Id int NOT NULL PRIMARY KEY,

Username nvarchar NOT NULL, Password nvarchar NOT NULL

);

# Lenguaje de Manipulación de Datos (DML)

*-- EFMigrationsHistory –*

+

Claro, aquí tienes cinco sentencias SQL `INSERT INTO` para la tabla ` EFMigrationsHistory` con columnas

`MigrationId` y `ProductVersion`. Los valores son ejemplos realistas que podrían usarse en un contexto de migraciones de Entity Framework:

```sql

INSERT INTO EFMigrationsHistory (MigrationId, ProductVersion) VALUES ('20230315120000\_InitialCreate', '5.0.0'); INSERT INTO EFMigrationsHistory (MigrationId, ProductVersion) VALUES ('20230316130000\_AddProductsTable', '5.0.0');

INSERT INTO EFMigrationsHistory (MigrationId, ProductVersion) VALUES ('20230317140000\_AddOrdersTable', '5.0.0');

INSERT INTO EFMigrationsHistory (MigrationId, ProductVersion) VALUES ('20230318150000\_UpdateCustomerSchema', '5.0.1');

INSERT INTO EFMigrationsHistory (MigrationId, ProductVersion) VALUES ('20230319160000\_RemoveObsoleteColumns', '5.0.1');

```

Estos ejemplos reﬂejan un historial de migraciones en una aplicación que utiliza Entity Framework, con identificadores de migración que incluyen una marca de tiempo y una descripción de la migración. Las versiones de producto son versiones ficticias de un paquete de Entity Framework.

*-- Users --*

Aquí tienes cinco sentencias SQL `INSERT INTO` para la tabla `Users` con columnas `Id`, `Username` y `Password`, utilizando datos de ejemplo realistas:

```sql

INSERT INTO Users (Id, Username, Password) VALUES (1, 'john\_doe', 'P@ssw0rd123'); INSERT INTO Users (Id, Username, Password) VALUES (2, 'jane\_smith', 'S3cur3P@ss!');

INSERT INTO Users (Id, Username, Password) VALUES (3, 'michael\_brown', 'M1chaelB#2023'); INSERT INTO Users (Id, Username, Password) VALUES (4, 'emily\_jones', 'Em1lyJ0nes$'); INSERT INTO Users (Id, Username, Password) VALUES (5, 'david\_wilson', 'D@vidW!lson2023');

```

Recuerda que en un entorno real, las contraseñas deben ser almacenadas de manera segura, utilizando técnicas de hashing y salting, en lugar de texto plano.