NoVoãr

Nombre y apellidos		Álvaro Sánchez Bernal			
Ciclo	1º DAM O	**Módulo**	BBDD	**Año**	2020
Práctica		Práctica UD4: redacción. modelo entidad-relación, modelo relacional y sentencias SQL			
Proyecto		UchiProgram crea bbdd para NoVoãr, una asociación portuguesa que investiga pingüinos en la Antártida			

ÍNDICE

Documentación de las prácticas anteriores

Toma de requisitos

From: NoVoãr To: UchiProgram 24/10/19 - v 1.01

Base de datos de investigación pingüinil

Queremos tener una base de datos en la que se consulte los pingüinos y sus características, además de dónde viven y a quién tienen a cargo.
Nos dividimos en equipos numerados formados por investigadores. Los investigadores son de dos clases, monitores (con equipos avanzados o no) o científicos (sabiendo el título de su tesis y campo de estudio). Nos gustaría describir brevemente cómo es cada equipo. Cada equipo tiene varios investigadores, pero solo un científico. De los investigadores queremos almacenar nombre, apellidos y dni.
Cada pingüino tiene asignado un monitor, y éste a su vez tiene asignado varios pingüinos que observar para realizar sus informes propios. Sin embargo, un pingüino puede ser examinado por muchos científicos y un científico examina a muchos pingüinos. Queremos conocer las fechas de los exámenes. De los pingüinos nos interesa conocer su sexo, nombre (si el monitor le ha puesto), y la localización de su colonia. Una vez que los pingüinos entren a una colonia, no podrán pertenecer a otra. Para identificar a los pingüinos usamos un código numérico único. Además, si el pingüino es un adulto. nos interesa saber cúal es su pareja (si tiene); y si es un polluelo, el estado de su muda.
También nos interesa almacenar los campamentos repartidos por las diferentes localizaciones del desierto helado, los cuales son formados por varios equipos. Los equipos sin embargo, sólo pertenecen a un campamento. De los campamentos queremos conocer su localización y nombre.
Importante aclarar que los pingüinos son muy sensibles a los cambios de su entorno, así que en una localización en concreto siempre se encuentra como máximo una colonia, la cual puede ser descrita en un campo añadido si así se desea.
Gracias por adelantado. Infórmanos sobre cualquier avance.

Atentamente,
NoVoãr

Cambios en la redacción de requisitos

FECHA	DESCRIPCIÓN
17/04/20	Como se ha normalizado la base de datos, nuevas tablas han sido agregadas. Ahora científico ha visto modificado su campo *campo_Estudio*, que ha pasado a llamarse *titulo_Tesis*. Por ello se ha cambiado en la redacción

Modelo Entidad-Relación

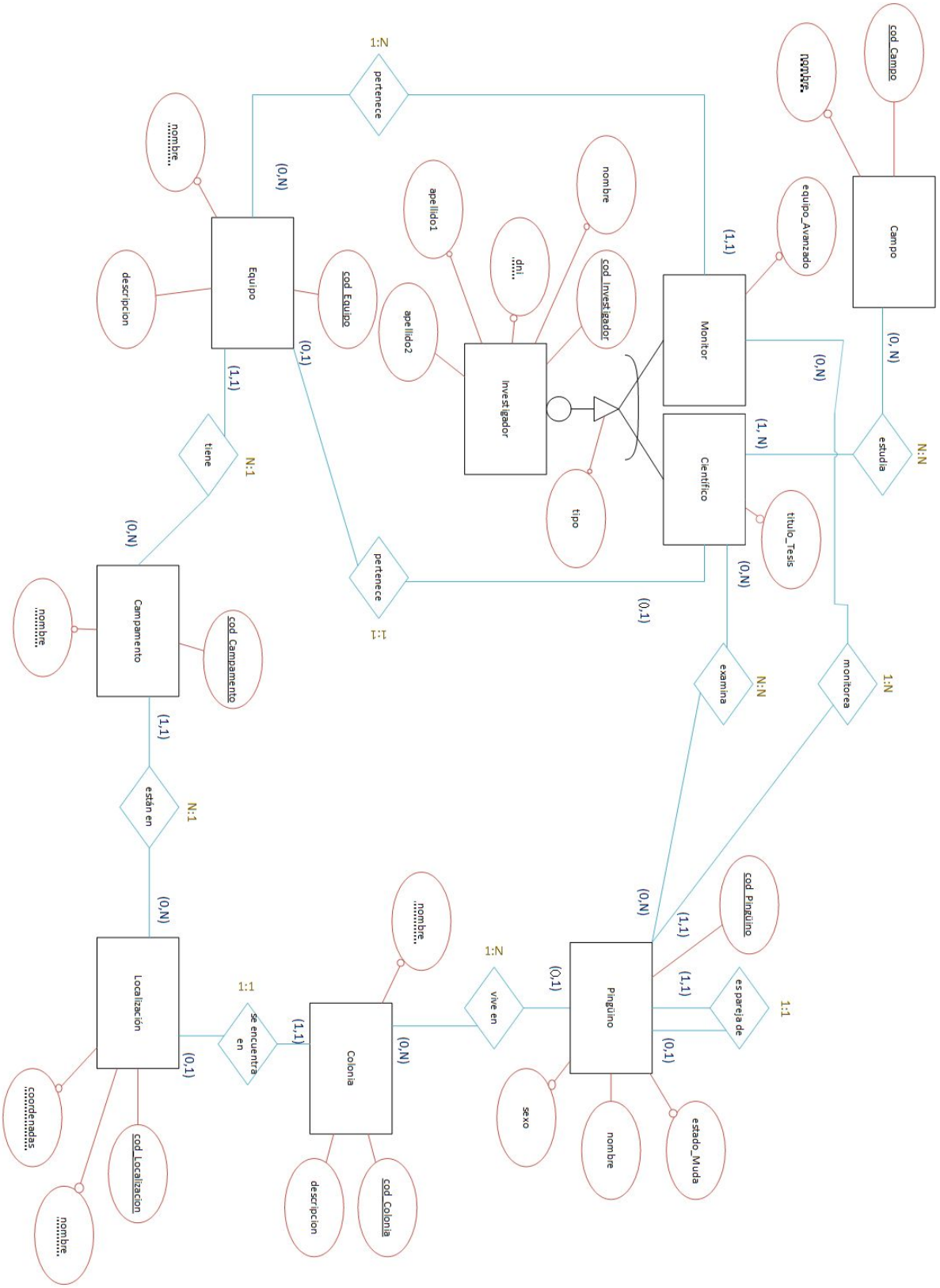


Tabla de dominios

Entidad	Atributo	Dominio
Investigador	cod_Investigador	Numérico
	nombre	Cadena car. (15)
	dni	Numérico (8)
	apellido1	Cadena car. (25)
	apellido2	Cadena car. (20)
Monitor	equipo_Avanzado	Booleano (Sí/No)
Científico	titulo_Tesis	Cadena car. (15)
Equipo	cod_Equipo	Numérico
	nombre	Cadena car. (8)
	descripción	Texto largo (100)
Campamento	cod_Campamento	Numérico
	nombre	Cadena car. (8)
Localización	cod_Localización	Numérico
	nombre	Cadena car. (20)
	coordenadas	(x, y) Entero (15)
Colonia	cod_Colonia	Numérico
	descripción	Texto largo (100)
	nombre	Cadena car. (20)
Pingüino	cod_Pingüino	Numérico
	nombre	Cadena car. (10)
	sexo	Carácter
	estado_Muda	Cadena car. (12)
Campo	cod_Campo	Numérico
	nombre	Cadena car. (15)

Aclaraciones

Pese a que no se especifica, debido a la definición de *colonia de pingüinos*, la participación de esta es (1,N) ya que en cada una hay numerosos pingüinos. También, hemos añadido un código de identificación en algunas entidades, bien para facilitar su búsqueda o porque así lo requerían.

Cambios en el modelo Entidad-Relación

Fecha	Práctica	Ubicación del cambio	Problema encontrado	Resolución
04/11/2019	P1	Entidad Colonia	No disponía de nombre y su participación era total (por lo que no se podía agregar sin pingüinos)	Se ha añadido un atributo único y no nulo llamado nombre y se ha cambiado a participación parcial
04/11/2019	P1	Entidad Pingüino	El atributo de uno de los subtipos era erróneo ya que se trataba de una relación recursiva.	Se ha cambiado la especialización por recursividad, ya que el atributo *pareja* hacía referencia a la CP de Pingüino. También se ha omitido el subtipo *Polluelo* para hacerlo más sencillo
04/11/2019	P1	Entidad Investigador	Los subtipos carecían de atributos, lo que lo hacía extraño	Se han añadido atributos convenientes para ambos subtipos
04/11/2019	P1	Entidad Equipo	Las participaciones totales impiden	Se han convertido en parciales

			la agregación de instancias a la base de datos	
04/11/2019	P1	Entidad Campapento	La participación total impide la agregación de instancias a la base de datos	Se ha convertido en parcial
04/11/2019	P1	Relación examina	-	Se ha agregado un atributo llamado fecha_Examen para tener registro de cuando se hizo.
06/02/2020	P1	Relación pertenece	Para agregar un equipo era necesario tener un científico sí o sí, por lo que daría error.	Se ha cambiado la participación mínima de Científico a 0
06/02/2020	P1	Relación se encuentra en	Para agregar una localización era necesario tener una colonia obligatoriamente	Se ha cambiado la participación mínima de Colonia a 0
17/04/2020	P4	Campo campo_Estudio	Se ha normalizado ese campo y no era necesario	Se ha sustituido por titulo_Tesis
17/04/2020	P4	Científico	campo_Estudio se ha normalizado	Se ha creado una nueva Entidad que se relaciona con Científico

Modelo Relacional

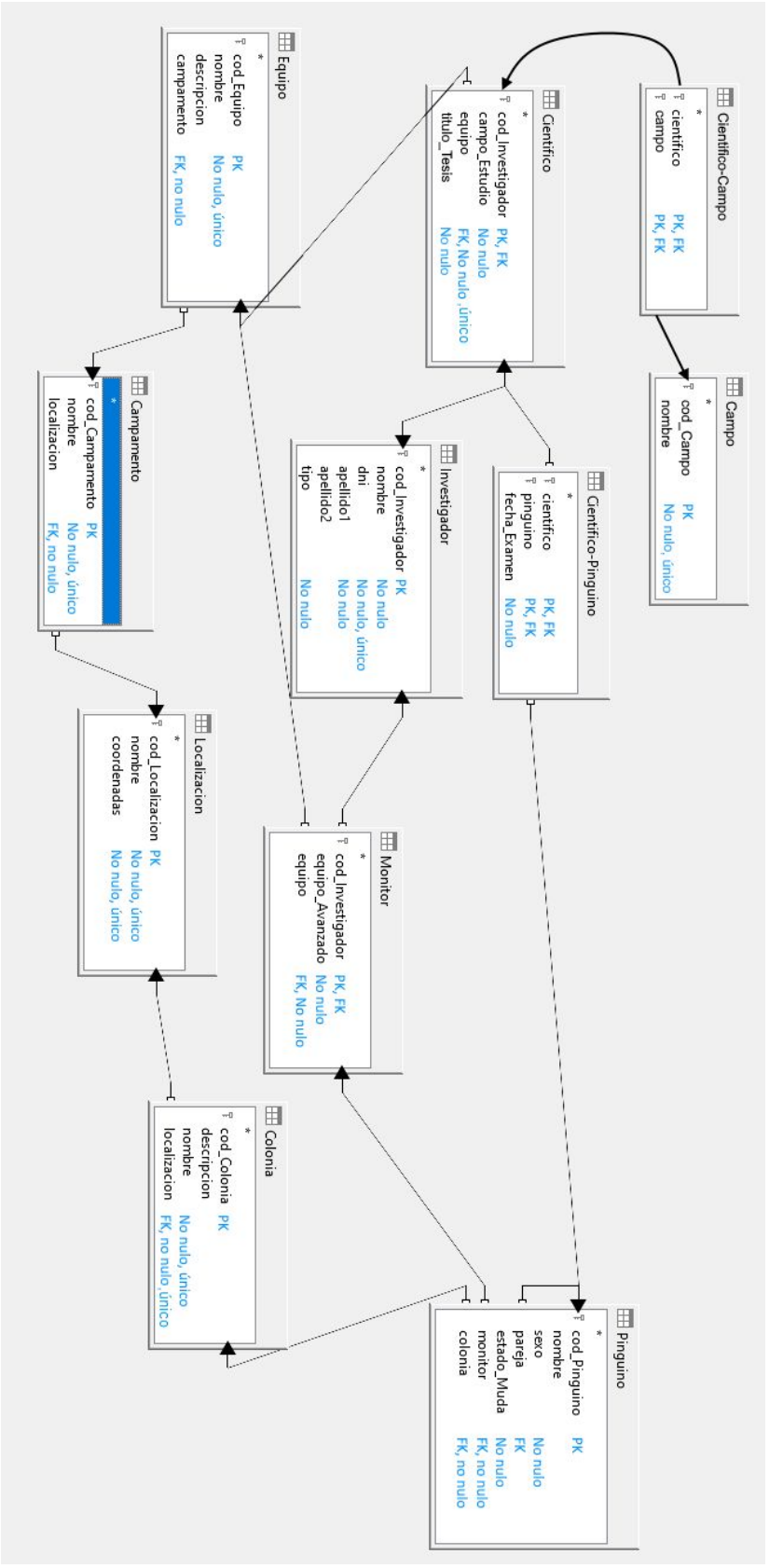


Tabla de dominios

Tabla	Atributo	Dominio
Investigador	cod_Investigador	Numérico
	nombre	Cadena car. (15)
	dni	Numérico (8)
	apellido1	Cadena car. (25)
	apellido2	Cadena car. (20)
	tipo	Numérico (1)
Monitor	cod_Investigador	Numérico
	equipo_Avanzado	Booleano (Sí/No)
	equipo	Numérico
Científico	cod_Investigador	Numérico
	titulo_Tesis	Cadena car. (15)
	equipo	Numérico
Equipo	cod_Equipo	Numérico
	nombre	Cadena car. (8)
	descripción	Texto largo (100)
	campamento	Numérico
Campamento	cod_Campamento	Numérico
	nombre	Cadena car. (8)
	localizacion	Numérico
Localización	cod_Localización	Numérico
	nombre	Cadena car. (20)
	coordenadas	(x, y) Entero (15)
Colonia	cod_Colonia	Numérico

	descripción	Texto largo (100)
	nombre	Cadena car. (20)
	localizacion	Numérico
Pingüino	cod_Pingüino	Numérico
	nombre	Cadena car. (10)
	sexo	Carácter
	estado_Muda	Cadena car. (12)
	pareja	Numérico
	monitor	Numérico
	colonia	Numérico
Cientifico-Campo	cientifico	Numérico
	campo	Numérico
Campo	cod_Campo	Numérico
	nombre	Cadena car. (15)

Aclaraciones

El atributo *equipo_Avanzado* en ***Monitor*** es un booleano. Esto es así para saber si el monitor dispone de un equipo de monitoreo avanzado o sigue usando la versión anterior.

Científico-Pingüino equivale a la relación *examina*

Cambios en el modelo Relacional

Fecha	Práctica	Ubicación del cambio	Problema encontrado	Resolución
06/02/2020	P2	Tabla Científico	En la clave ajena equipo no figuraba la restricción de único	Se ha agregado
06/02/2020	P2	Tabla Colonia	En la clave ajena de	Se ha agregado

			localización no figuraba la restricción de unicidad	
17/04/2020	P4	Tabla Científico	Se ha normalizado campo_Estudio	Se han agregado dos nuevas tablas relacionadas con Científico para cumplir este propósito

Normalización

Para este proyecto se ha considerado normalizar al Investigador de tipo Científico. Antes tenía un campo llamado campo_Estudio que representaba en que campo trabajaba el científico, pero sólo dejaba introducir uno, lo cual no tenía mucho sentido. Por ello, se ha decidido cambiarlo por tablas con una relación muchos a muchos, ya que un científico puede ser experto en más de un campo. También se ha agregado un campo llamado tiulo_Tesis a la entidad Científico para adquirir más información sobre él.

Planificación

A continuación se muestra una tabla con el orden de creación de éstas:

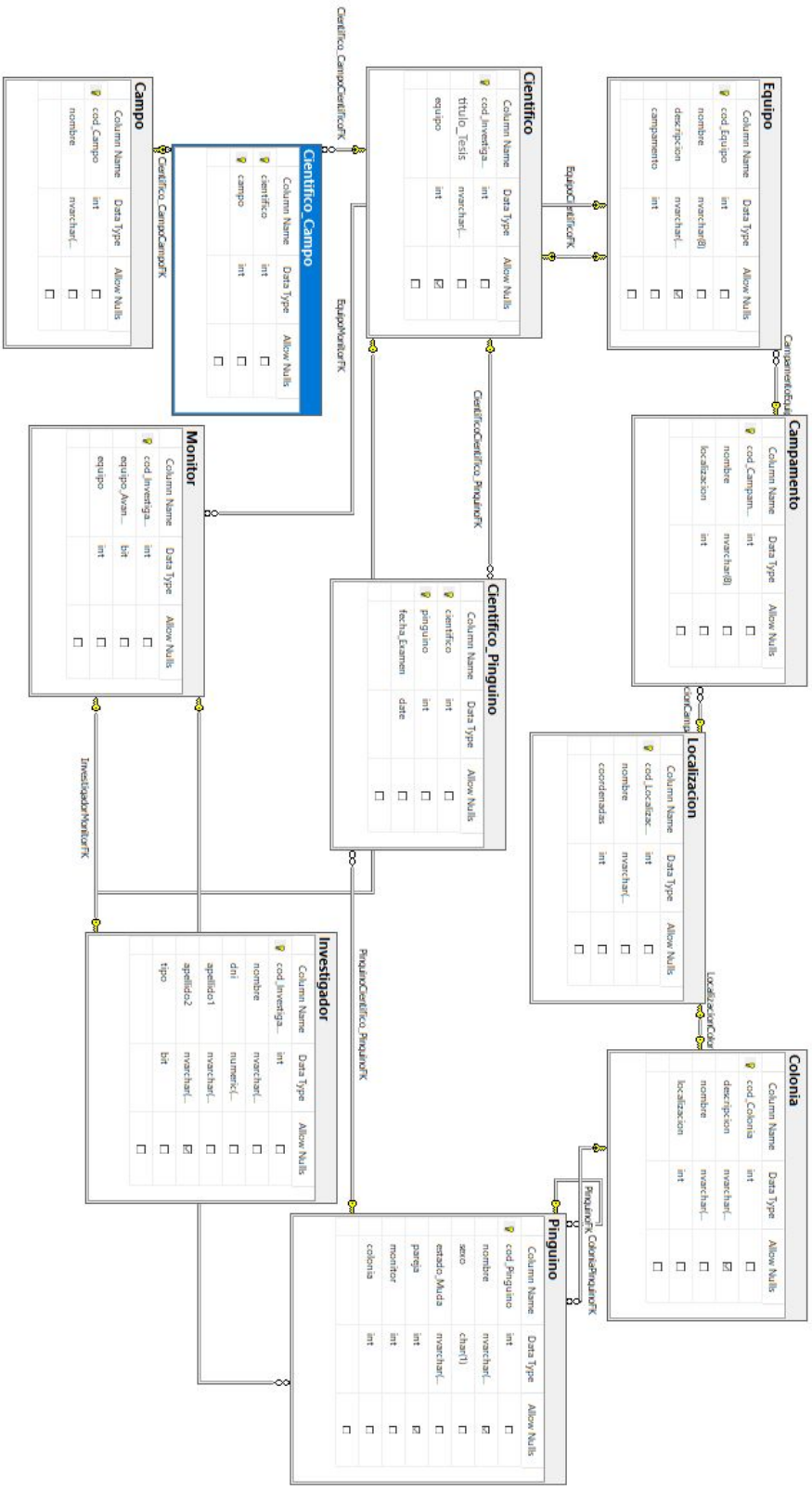
Orden de creación	Nombre de la tabla
1	Localización
2	Investigador
3	Campamento
4	Colonia
5	Equipo
6	Científico
7	Monitor
8	Pingüino
9	Científico-Pingüino
10	Campo
11	Científico-Campo

Tipos de datos

TIPO DE DATO	TABLA-COLUMNA	EXPLICACIÓN
int identity	Todas-Tabla de código	Usamos esto porque este tipo de dato automático es muy cómodo para trabajar.
date	Cientifico_Pingüino-fecha_Examen	Este tipo de dato es necesario para guardarlo como lo que es, una fecha
nvarchar	Todas-nombre	Usado con un limitador de longitud ya que los nombres pueden tener diferentes longitudes
bit	Monitor y Científico-tipo y equipo_Avanzado	Ahorramos espacio ya que esto son simplemente booleanos
char	Pinguino-sexo	Solo tiene una longitud de 1 debido a que es una letra (M ó H)
int	Todas-Claves foráneas	El valor de las claves foráneas

		debe ser del mismo tipo que el de las primarias en sus respectivas tablas

Diagrama relacional de MMS



Cambios en la planificación

Fecha	Práctica	Ubicación del cambio	Problema encontrado	Resolución
17/04/2020	P4	Tabla de orden de creación de las tablas	Carecían de las nuevas tablas agregadas por la normalización	Se han agregado ambas al plan de creación
17/04/2020	P4	Diagrama de MMS	No estaban las nuevas tablas agregadas por la normalización	Se han agregado

Operaciones CRUD

Tabla	Nº operación	Descripción
Pingüino	1	Crear un nuevo pingüino, enlazarlo a la colonia con PK 1, enlazarlo al monitor con PK 2, no tendrá pareja
	2	Cambiar nombre del pingüino con PK 3
	3	Contar a las pingüinas que tienen pareja
	4	Buscar a los pingüinos a los que han realizado un examen
	5	Eliminar a los pingüinos cuyo estado de muda sea “Infantil”
Colonia	1	Cuenta los pingüinos que hay en la colonia llamada “Elephant”
	2	Modifica la descripción de la colonia con PK 1
	3	Muestra el nombre de las colonias que tengan algún pingüino llamado “Willy”
	4	Busca colonia/localización/campamento atendiendo a los valores de colonia
	5	Modifica el nombre de la colonia con PK 1 a “Tuxedo”
Localización	1	Contar las colonias que hay en la localización con nombre “Puerto Lockroy”
	2	Cuenta todas las localizaciones actuales
	3	Cambia el nombre de la localización con PK 2 a “Puerto Charcot”
	4	Busca la localización con coordenadas 115956832
	5	Busca localización/colonias/campamentos en base al valor de localización
Campamento	1	Agregar un nuevo campamento, enlazarlo con la localización con PK 2, su nombre

		debe ser “Uranus”
	2	Agregar un nuevo campamento, enlazarlo con la localización con PK 1, su nombre debe ser “Titan”
	3	Contar cuántos campamentos existen actualmente
	4	Buscar campamentos/equipos/investigadores en base al campamento
	5	Buscar campamento/localización/colonia en base al campamento
Equipo	1	Agregar un nuevo equipo con nombre “hScott”, descripción a placer y enlazarlo con el campamento con PK 1
	2	Contar los investigadores que tiene el equipo “tNova” (PK 1)
	3	Modificar el nombre del equipo con PK 2 a “dScott”
	4	Eliminar el equipo con PK 2
	5	Buscar equipo/investigador/científico/científico-pingüino/pingüino en base al equipo

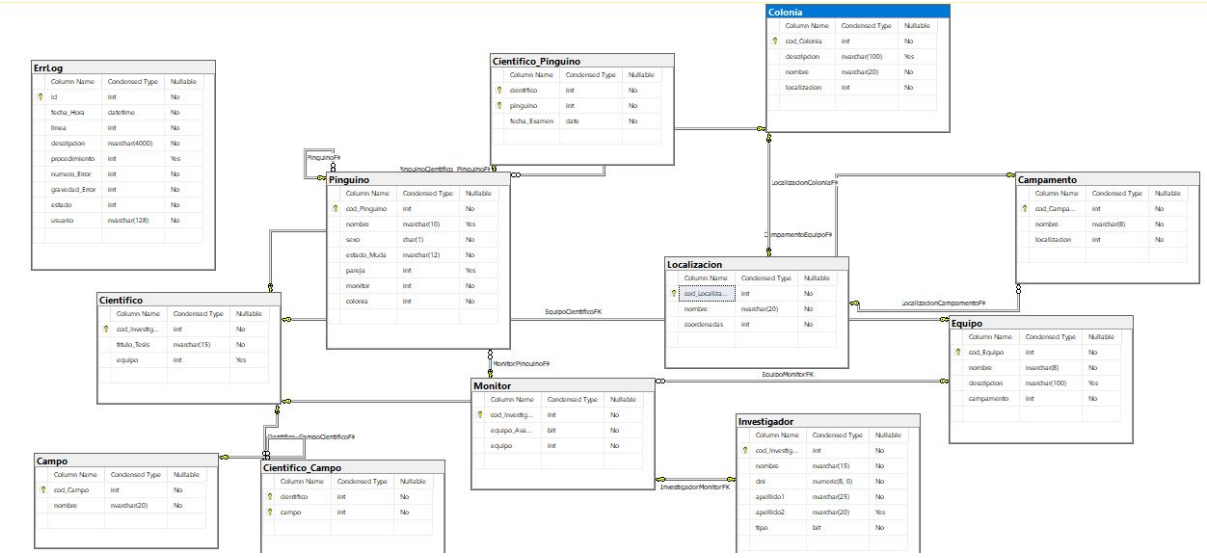
Compañero: Manuel Martínez Castedo

Las operaciones CRUD de Manuel se encuentran en la carpeta con nombre “**Operaciones CRUD de Manuel Martínez Castedo**”. También he dejado algunos datos que puedes insertar en mi base de datos para facilitar el trabajo en “**(Para facilitar) Inserción de algunos datos en la bbdd de Álvaro**”.

Control y monitorización de errores

En este apartado trataremos las transacciones y el control de errores.

Tabla ErrLog



Esta nueva tabla almacenará los errores que se produzcan en las operaciones CRUD al modificar la base de datos.

También disponemos de una vista que permitirá ver los errores que se han producido en el día de hoy.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [id]
      ,[fecha_Hora]
      ,[linea]
      ,[descripcion]
      ,[procedimiento]
      ,[numero_Error]
      ,[gravedad_Error]
      ,[estado]
      ,[usuario]
  FROM [DAM_SANCHEZ_BERNAL_ALVARO].[dbo].[vErroresHoy]
```

Results | Messages

id	fecha_Hora	linea	descripcion	procedimiento	numero_Error	gravedad_Error	estado	usuario
5	2020-05-17 13:49:19.203	3	Infracción de la restricción PRIMARY KEY 'Cien...	NULL	2627	14	1	dbo
7	2020-05-17 13:50:07.940	3	Infracción de la restricción PRIMARY KEY 'Cien...	NULL	2627	14	1	dbo
9	2020-05-17 13:50:08.690	3	Infracción de la restricción PRIMARY KEY 'Cien...	NULL	2627	14	1	dbo
11	2020-05-17 13:50:09.213	3	Infracción de la restricción PRIMARY KEY 'Cien...	NULL	2627	14	1	dbo

Demostración del funcionamiento

Para las operaciones de selección, funciona de la siguiente manera:

```
DECLARE @cod_equipo INT = 1;
BEGIN TRY
    IF @cod_equipo IS NULL
        RAISERROR('El valor de cod_equipo no es correcto', 16, 1);
    ELSE
        select *
        from Equipo as E
        inner join
        Cientifico as C
        ON
        C.equipo = E.cod_Equipo
        INNER JOIN
        Investigador AS I
```

Primero se declara la variable con lo que interesa buscar

```
select *
from Equipo as E
inner join
Cientifico as C
ON
C.equipo = E.cod_Equipo
INNER JOIN
Investigador AS I
ON
I.cod_Investigador = C.cod_Investigador
INNER JOIN
Cientifico_Pinguino AS CP
ON
CP.cientifico = C.cod_Investigador
INNER JOIN
```

Si el resultado es correcto, se ejecuta la sentencia SQL

cod_Equipo	nombre	descripcion	campamento	cod_Investigador	titulo_Tesis	equipo	cod_Investigador	nombre	dni	apellido1	apellido2	tipo	cientifico
1	tNova	Equipo especializado en mantenimiento de instala...	1	1	Vida sob o gelo	1	1	Camilo	12345678	Alves	Belo	1	1

Por ende, se muestra la tabla con el resultado… Veamos ahora qué ocurre cuando los valores introducidos no son válidos (o son nulos)

```
DECLARE @cod_equipo INT;
BEGIN TRY
    IF @cod_equipo IS NULL
        RAISERROR('El valor de cod_equipo no es correcto', 16, 1);
    ELSE
        select *
        from Equipo as E
        inner join
```

Aquí ejecutamos la misma sentencia, pero esta vez la variable de cod_equipo tiene un valor nulo, por lo que dará error.

```
DECLARE @cod_equipo INT;
BEGIN TRY
    IF @cod_equipo IS NULL
        RAISERROR('El valor de cod_equipo no es correcto', 16, 1);
    ELSE
```

El IF dará true, por lo que se lanzará la excepción

```
        CP.pinguino = P.cod_Pinguino
        WHERE E.cod_Equipo = @cod_equipo;
END TRY
BEGIN CATCH
    PRINT 'HA OCURRIDO UN ERROR. MIRA vErroresHOY PARA VER MÁS DETALLES'
    EXECUTE almacenar_error
END CATCH;
```

Se mostrará en la consola un mensaje de error y se almacenará el error

```
CREATE PROCEDURE [dbo].[almacenar_error]
AS
    INSERT INTO ErrLog(fecha_Hora, linea, descripcion,
                       procedimiento, numero_Error,
                       gravedad_Error, estado, usuario
    VALUES (GETDATE(), ERROR_LINE(), ERROR_MESSAGE(),
            ERROR_PROCEDURE(), ERROR_NUMBER(),
            ERROR_SEVERITY(), ERROR_STATE(), USER);
```

Almacenar error se encarga de insertar los valores correspondientes en la tabla ErrLog

44	59	2020-05-17 16:45:49.343	5	El valor de cod_equipo no es correcto	NULL	50000	16	1	dbo

Por último, al dirigirnos a la tabla vErroresHoy, podremos ver como el error se ha almacenado.

Veamos ahora el caso en un UPDATE:

```
BEGIN TRY
    BEGIN TRANSACTION T3
        UPDATE
            Pinguino
        SET
            nombre='Felipe'
        WHERE
            cod_Pinguino=3;
    COMMIT TRANSACTION T3
    PRINT 'Operación realizada con éxito'
END TRY
```

Se desea cambiar el nombre del Pinguino con codigo 3 a Felipe

```
BEGIN TRY
    BEGIN TRANSACTION T3
        UPDATE
            Pinguino
        SET
            nombre='Felipe'
        WHERE
            cod_Pinguino=3;
    COMMIT TRANSACTION T3
    PRINT 'Operación realizada con éxito'
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION T3
    PRINT 'HA OCURRIDO UN ERROR. MIRA vErroresHOY PARA VER MÁS DETALLES'
```

Se ejecuta la sentencia y se realiza con éxito.

3	Felipe	M	Infantil	NULL	2	1

Y como podemos ver, el pingüino ahora tiene el nombre Felipe

Vamos a ver qué ocurre cuando hay un error:

```
BEGIN TRY
    BEGIN TRANSACTION T3
        UPDATE
            Pinguino
        SET
            nombre='Antonio'
        WHERE
            cod_Pinguino='hola que tal';
    COMMIT TRANSACTION T3
    PRINT 'Operación realizada con éxito'
END TRY
BEGIN CATCH
```

Esta vez, el codigo tiene un valor que no es correcto

```
BEGIN TRY
    BEGIN TRANSACTION T3
        UPDATE
            Pinguino
        SET
            nombre='Antonio'
        WHERE
            cod_Pinguino='hola que tal';
    COMMIT TRANSACTION T3
    PRINT 'Operación realizada con éxito'
END TRY
```

Se ejecuta la sentencia SQL

```
            nombre='Antonio'
        WHERE
            cod_Pinguino='hola que tal';
    COMMIT TRANSACTION T3
    PRINT 'Operación realizada con éxito'
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION T3
    PRINT 'HA OCURRIDO UN ERROR. MIRA vErroresHOY PARA VER MÁS DETALLES'
    EXECUTE almacenar_error
END CATCH;
```

Pero como es de esperar, lanza un error ya que la clave primaria no tiene el valor correcto

```
CREATE PROCEDURE [dbo].[almacenar_error]
AS
    INSERT INTO ErrLog(fecha_Hora, linea, descripcion,
                       procedimiento, numero_Error,
                       gravedad_Error, estado, usuario)
    VALUES (GETDATE(), ERROR_LINE(), ERROR_MESSAGE(),
            ERROR_PROCEDURE(), ERROR_NUMBER(),
            ERROR_SEVERITY(), ERROR_STATE(), USER);
```

Así se ejecuta almacenar_error

61	2020-05-18 15:01:40.707	4	Error de conversión al convertir el valor varchar 'h...	NULL	245	16	1	dbo

Y como podemos comprobar, el error queda guardado en ErrLog