

SPRINT 3

Diseño y construcción de la aplicación 1MASQR TUR

PRESENTADO POR: Eliel Mauricio Escobar Victoria

PRESENTADO A: Universidad Tecnológica de Pereira - UTP

OCTUBRE 10 de 2022

ESPECIFICACIÓN DEL MODELO CONCEPTUAL DEL SISTEMA DE INFORMACIÓN A DESARROLLAR

SPRINT 3. Elaborar el modelo del sistema de información a desarrollar, teniendo en cuenta modelos de desarrollo de software vistos hasta el momento con los profesores de MisionTic de la UTP; así mismo los requerimientos del cliente, normas, protocolos y modelos de calidad en el desarrollo de sistemas de información.

DIAGRAMA ENTIDAD RELACION APLICACION 1MASQR TUR (MER)

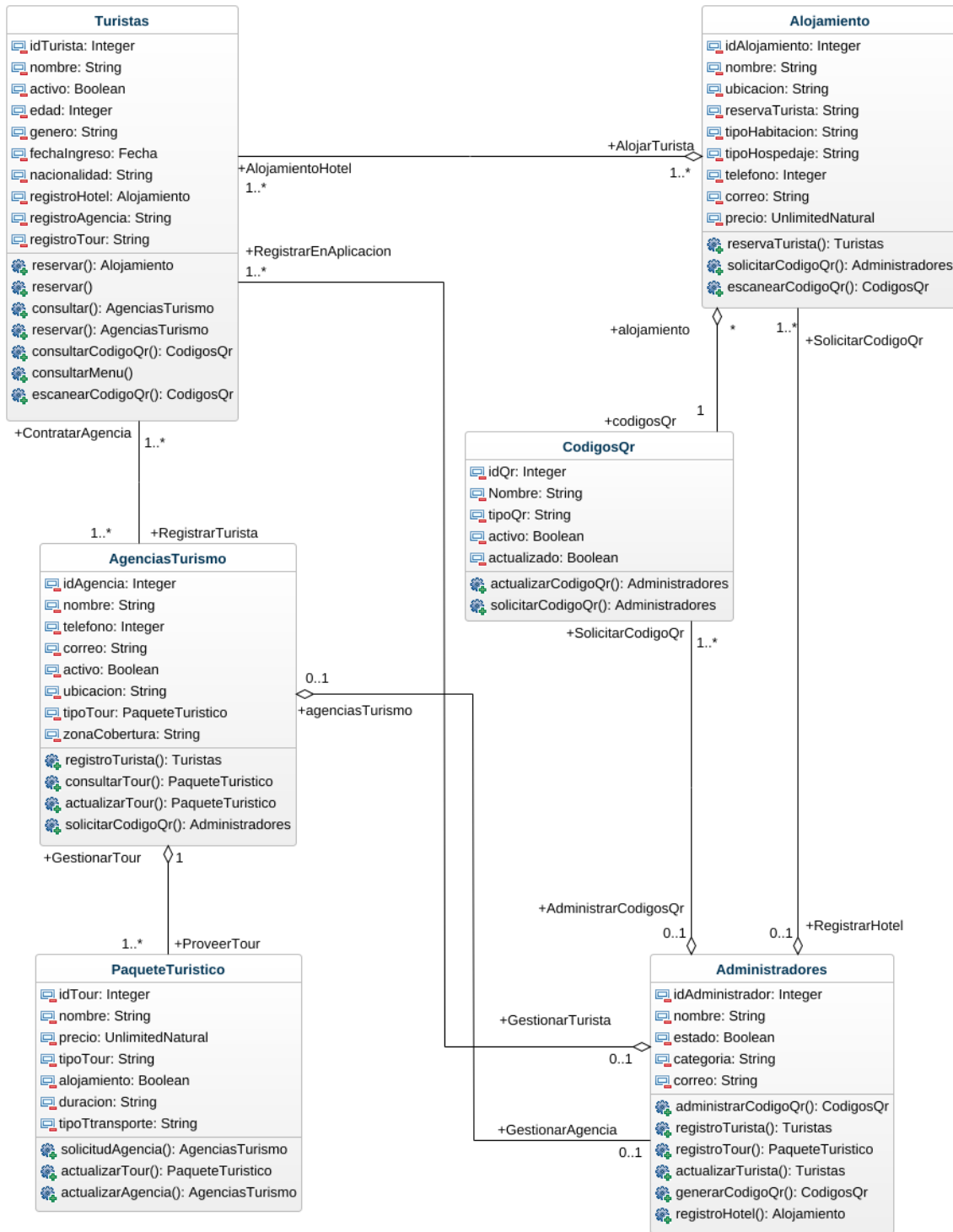
MARCO TEORICO Y CONTEXTUAL

Las bases de datos tienen como fin almacenar información y esos datos son almacenados de forma ordenada en tablas, por lo que una vez que has creado tu base de datos y las tablas, el siguiente paso es comenzar a guardar información, además de tener la opción para consultar la información que hemos almacenado.

Para la capa de lenguaje de definición de datos en primer lugar se creó el modelo entidad-relación de la base de datos 1masQR_Tur.. En segundo lugar, se creó un diagrama de base de datos, puesto que así es más eficiente crear las tablas para posteriormente relacionarlas entre sí. Posteriormente se creó cada una de las tablas requeridas para el modelo de entidad relación, el nombre de las tablas corresponde al mismo nombre de las entidades en el modelo entidad-relación y sus atributos en singular, dentro de los campos a llenar se asignó el nombre a la columna, tipo de datos y si se permitía valores nulos. Posteriormente se normalizaron las tablas y finalmente se asignó la llave principal a las tablas si era el caso y posteriormente se prestaba como llave foránea a otra tabla.

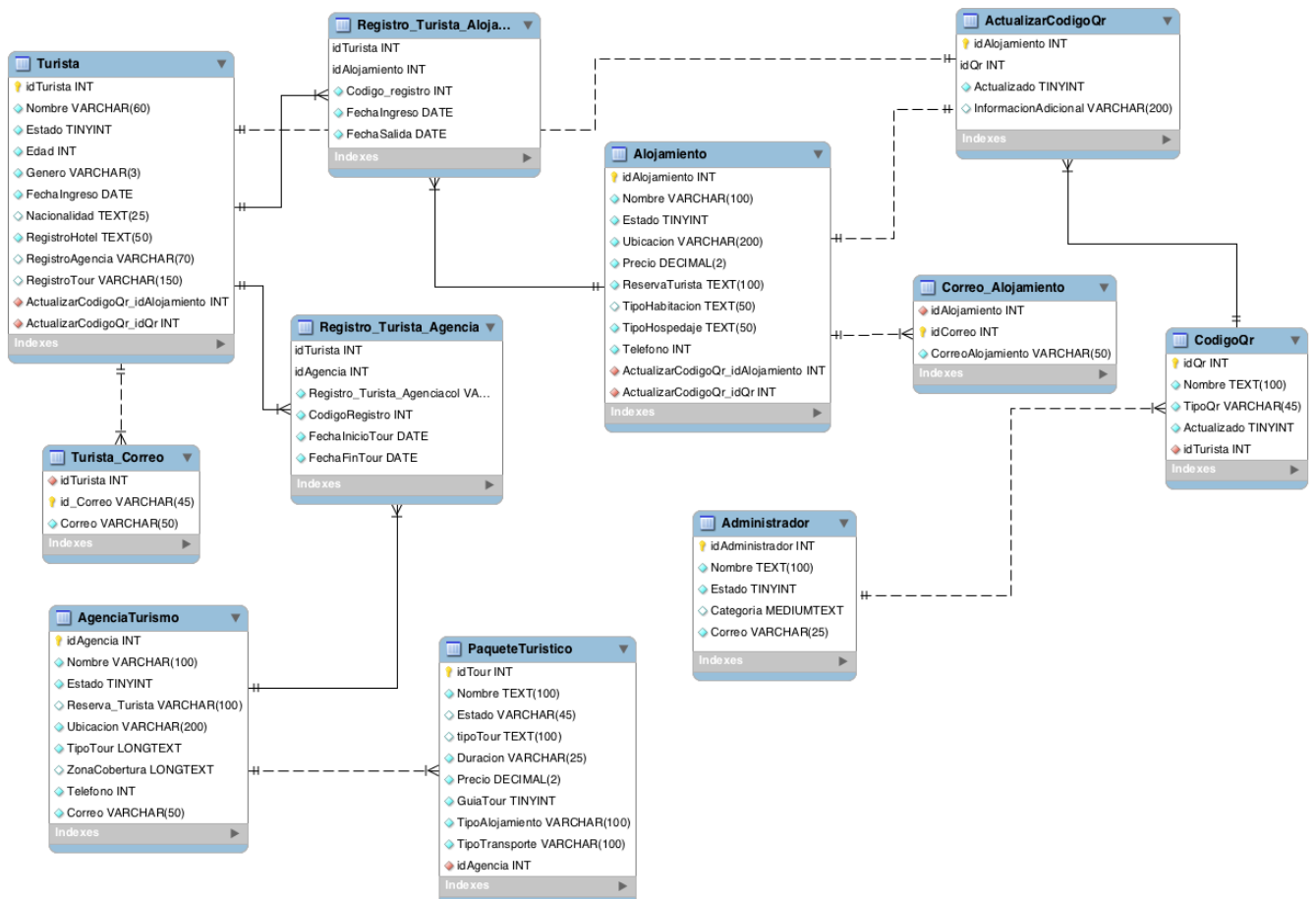
MODELO RELACIONAL CREADO CON LA HERRAMIENTA GENMYMODEL.

CLASES, OBJETOS, ATRIBUTOS, METODOS, RELACIONES.



Elaboración del modelo de procesos o diagrama de actividades: una vez determinado el diagrama de clases se debe modelar los procesos a través de los diagramas de secuencias, diagramas de colaboración y diagramas de actividad. Otro de los diagramas como el denominado MER (Diagrama Entidad-Relación) de actividades permite mostrar la secuencia

de las acciones de varios objetos y seleccionar el orden en que se harán las cosas. Representa los procesos de negocios, incluidos el flujo de datos. También puede utilizarse para modelar lógica compleja y/o paralela dentro de un sistema. Un diagrama MER es la versión UML de un diagrama de flujo y se usan para analizar procesos.



DICCIONARIO DE DATOS DE 1masQR Tur

Tabla_01		Turista		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idAgencia	INT		Primaria
2	Nombre	VARCHAR	100	
3	Estado	BOOLEAN		
4	Edad	INT		
5	Genero	VARCHAR	3	
6	Fecha_Ingreso	DATE		
7	Nacionalidad	VARCHAR	25	
8	Registro_Hotel	VARCHAR	50	
9	Registro_Agencia	VARCHAR	70	
10	Registro_Tour	VARCHAR	150	
11	idAdministrador	INT		Foranea

Tabla_02		Alojamiento		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idAlojamiento	INT		Primaria
2	Nombre	VARCHAR	100	
3	Estado	BOOLEAN		
4	Ubicación	INT		
5	Precio	DECIMAL(2)	2	
6	ReservaTurista	TEXT(100)	100	
7	TipoHabitacion	TEXT(100)	50	
8	TipoHospedaje	TEXT(100)	50	
9	Telefono	INT		
10	Correo	TEXT	30	
11	idAdministrador	INT		Foranea

Tabla_03		AgenciaTurismo		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idAgencia	INT		Primaria
2	Nombre	VARCHAR	100	
3	Estado	BOOLEAN		
4	Ubicación	VARCHAR	200	
5	TipoTour	LONGTEXT	2	
6	ReservaTurista	VARCHAR	100	
7	ZonaCobertura	TEXT(100)	50	
8	Telefono	INT		
9	Correo	TEXT	30	
10	idAdministrador	INT		Foranea
11	Id_Turista	INT		Foranea

Tabla_04		PaqueteTuristico		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idTour	INT		Primaria
2	Nombre	VARCHAR	100	
3	Estado	BOOLEAN		
4	TipoTour	VARCHAR	200	
5	Duracion	VARCHAR	25	
6	Precio	DECIMAL	2	
7	GuiaTour	BOOLEAN		
8	TipoAlojamiento	VARCHAR	100	
9	TipoTransporte	VARCHAR	100	
10	idAgencia	INT		Foranea
11	idTurista	INT		Foranea

Tabla_05		Administrador		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idAdministrador	INT		Primaria

2	Nombre	VARCHAR	100	
3	Estado	BOOLEAN		
4	Categoria	MEDIUMTEXT		
5	Correo	VARCHAR	25	

Tabla_06		Correo_Turista		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idTurista	INT		Foranea
2	idCorreo	VARCHAR	100	
3	Correo	VARCHAR	50	

Tabla_07		Registro_Turista_Alojamiento		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idTurista	INT		Primaria
2	idAlojamiento	INT		Primaria
3	CodigoRegistro	INT		
4	FechaIngreso	DATE		
5	FechaSalida	DATE		

Tabla_08		Registro_Turista_Agencia		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idTurista	INT		Primaria
2	idAgencia	INT		Primaria
3	Registro_Turista	VARCHAR	100	
4	CodigoRegistro	INT		
5	FechaInicioTour	DATE		
6	FechaFinTour	DATE		

Tabla_09		Correo_Alojamiento		
No	Nombre Campo	Tipo	Longitud	Tipo llave
1	idAlojamiento	INT		Foranea
2	idCorreo	INT		
3	CorreoAlojamiento	VARCHAR	50	

SENTENCIAS SQL PARA CREAR LAS TABLAS DE LA BASE DE DATOS 1MASQR

-- MySQL Workbench Forward Engineering --

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,

SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-- Schema 1masQR_Tur

-- Schema 1masQR_Tur

```
CREATE SCHEMA IF NOT EXISTS `1masQR_Tur` DEFAULT CHARACTER SET utf8 ;
USE `1masQR_Tur` ;
```

-- Table `1masQR_Tur`.`Administrador`

```
CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`Administrador` (
  `idAdministrador` INT NOT NULL COMMENT 'Llave primaria, codigo unico del
Administrador',
  `Nombre` TEXT(100) NOT NULL COMMENT 'Nombre del Administrador del sistema',
  `Estado` TINYINT NOT NULL COMMENT 'Si se encuentra activo o no.',
  `Categoria` MEDIUMTEXT NULL COMMENT 'Categoria del administrador (principal o
secundario)',
  `Correo` VARCHAR(25) NOT NULL COMMENT 'Correo del administrador del sistema',
  PRIMARY KEY (`idAdministrador`))
ENGINE = InnoDB;
```

-- Table `1masQR_Tur`.`CodigoQr`

```
CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`CodigoQr` (
  `idQr` INT NOT NULL COMMENT 'Llave primaria, unica para cada código QR ',
  `Nombre` TEXT(100) NOT NULL COMMENT 'Nombre de cada código qr, dependiendo del
tipo ',
  `TipoQr` VARCHAR(45) NOT NULL COMMENT 'Tipo de código QR (estatico o dinamico)',
  `Actualizado` TINYINT NOT NULL COMMENT 'Si se encuentra actualizado o no',
  `idTurista` INT NOT NULL COMMENT 'Id del administrador (llave foranea)',
  PRIMARY KEY (`idQr`),
  INDEX `fk_CodigoQr_Administrador1_idx` (`idTurista` ASC),
  CONSTRAINT `fk_CodigoQr_Administrador1`
    FOREIGN KEY (`idTurista`)
      REFERENCES `1masQR_Tur`.`Administrador` (`idAdministrador`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

-- Table `1masQR_Tur`.`ActualizarCodigoQr`

```
CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`ActualizarCodigoQr` (
  `idAlojamiento` INT NOT NULL COMMENT 'Llave primaria, unica para el Hotel',
  `idQr` INT NOT NULL COMMENT 'Nombre de cada código qr, dependiendo del tipo ',
  `Actualizado` TINYINT NOT NULL COMMENT 'Si se encuentra actualizado o no',
```

```

`InformacionAdicional` VARCHAR(200) NULL COMMENT 'Información adicional asociada
con el código QR o con información adicional del alojamiento solicitado',
PRIMARY KEY (`idAlojamiento`, `idQr`),
INDEX `fk_Alojamiento_has_CodigoQr_CodigoQr1_idx` (`idQr` ASC),
CONSTRAINT `fk_Alojamiento_has_CodigoQr_CodigoQr1`
FOREIGN KEY (`idQr`)
REFERENCES `1masQR_Tur`.`CodigoQr` (`idQr`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `1masQR_Tur`.`Turista`
-----

```

```

CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`Turista` (
  `idTurista` INT NOT NULL COMMENT 'Identificador unico para cada turista registrado en el
sistema',
  `Nombre` VARCHAR(60) NOT NULL COMMENT 'Nombre del usuario',
  `Estado` TINYINT NOT NULL COMMENT 'Activo (falso o verdadero)',
  `Edad` INT NOT NULL COMMENT 'Edad del usuario registrado',
  `Genero` VARCHAR(3) NOT NULL COMMENT 'Genero del usuario registrado',
  `FechaIngreso` DATE NOT NULL COMMENT 'Fecha de ingreso o registro del usuario en el
sistema',
  `Nacionalidad` TEXT(25) NULL COMMENT 'Nacionalidad el usuario registrado',
  `RegistroHotel` TEXT(50) NOT NULL COMMENT 'Registro del Hotel u otro tipo de hospedaje
escogido por el usuario',
  `RegistroAgencia` VARCHAR(70) NULL COMMENT 'Registro de la agencia de viajes escogida
por el usuario',
  `RegistroTour` VARCHAR(150) NULL COMMENT 'Registro del paquete turistico o tour
turistico escogido por el usuario',
  `ActualizarCodigoQr_idAlojamiento` INT NOT NULL,
  `ActualizarCodigoQr_idQr` INT NOT NULL,
PRIMARY KEY (`idTurista`),
INDEX `fk_Turista_01_ActualizarCodigoQr1_idx` (`ActualizarCodigoQr_idAlojamiento` ASC,
`ActualizarCodigoQr_idQr` ASC),
CONSTRAINT `fk_Turista_01_ActualizarCodigoQr1`
FOREIGN KEY (`ActualizarCodigoQr_idAlojamiento`, `ActualizarCodigoQr_idQr`)
REFERENCES `1masQR_Tur`.`ActualizarCodigoQr` (`idAlojamiento`, `idQr`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `1masQR_Tur`.`Alojamiento`
-----

```

```

CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`Alojamiento` (
  `idAlojamiento` INT NOT NULL COMMENT 'Llave primaria, unica para el Hotel',
  `Nombre` VARCHAR(100) NOT NULL COMMENT 'Nombre del Hotel o Alojamiento',
  `Estado` TINYINT NOT NULL COMMENT 'Si se encuentra activo o no.',

```



```

`Ubicacion` VARCHAR(200) NOT NULL COMMENT 'Ubicación del Hotel o Alojamiento,
contiene tipo de calle, barrio ',
`Precio` DECIMAL(2) NOT NULL COMMENT 'Precio del Alojamiento por Turista',
`ReservaTurista` TEXT(100) NOT NULL COMMENT 'Reserva del turista para utilizar los
servicios del Hotel u Hospedaje',
`TipoHabitacion` TEXT(50) NULL COMMENT 'Tipo de Habitacion que se va a contratar con el
usuario (doble, sencilla, etc.)',
`TipoHospedaje` TEXT(50) NOT NULL COMMENT 'Tipo de alojamiento disponible, hotel,
camping, etc',
`Telefono` INT NOT NULL COMMENT 'Telefono de contacto',
`ActualizarCodigoQr_idAlojamiento` INT NOT NULL,
`ActualizarCodigoQr_idQr` INT NOT NULL,
PRIMARY KEY (`idAlojamiento`),
INDEX `fk_Alojamiento_02_ActualizarCodigoQr1_idx` (`ActualizarCodigoQr_idAlojamiento`
ASC, `ActualizarCodigoQr_idQr` ASC),
CONSTRAINT `fk_Alojamiento_02_ActualizarCodigoQr1`
FOREIGN KEY (`ActualizarCodigoQr_idAlojamiento`, `ActualizarCodigoQr_idQr`)
REFERENCES `1masQR_Tur`.`ActualizarCodigoQr` (`idAlojamiento`, `idQr`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `1masQR_Tur`.`AgenciaTurismo`
-----

```

```

CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`AgenciaTurismo` (
`idAgencia` INT NOT NULL COMMENT 'Identificador unico de la agencia turistica (llave
primaria)',
`Nombre` VARCHAR(100) NOT NULL COMMENT 'Nombre de la agencia turística
registrada',
`Estado` TINYINT NOT NULL COMMENT 'Si se encuentra activa o no',
`Reserva_Turista` VARCHAR(100) NULL COMMENT 'Reserva hecha por el turista a la
agencia turistica ',
`Ubicacion` VARCHAR(200) NOT NULL COMMENT 'Dirección de la agencia turística
registrada',
`TipoTour` LONGTEXT NOT NULL COMMENT 'Tipo o categoría del tour contratado por el
usuario',
`ZonaCobertura` LONGTEXT NULL COMMENT 'Cobertura geográfica que tiene la agencia
turística para operar',
`Telefono` INT NOT NULL COMMENT 'Teléfono de contacto de la agencia turística',
`Correo` VARCHAR(50) NOT NULL COMMENT 'Correo unico de la agencia turistica',
PRIMARY KEY (`idAgencia`))
ENGINE = InnoDB;

```

```

-----
-- Table `1masQR_Tur`.`PaqueteTuristico`
-----

```

```

CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`PaqueteTuristico` (

```

```

`idTour` INT NOT NULL COMMENT 'Identificador unico del paquete turistico (llave
primaria)',
`Nombre` TEXT(100) NOT NULL COMMENT 'Nombre del paquete turistico',
`Estado` VARCHAR(45) NULL COMMENT 'Si se encuentra activo o no',
`tipoTour` TEXT(100) NULL COMMENT 'Tipo de paquete turistico',
`Duracion` VARCHAR(25) NOT NULL COMMENT 'Duración del tour o paquete turistico',
`Precio` DECIMAL(2) NOT NULL COMMENT 'Precio del tour o del paquete turistico',
`GuiaTour` TINYINT NOT NULL COMMENT 'Si cuenta con guía el recorrido o no.',
`TipoAlojamiento` VARCHAR(100) NOT NULL COMMENT 'Tipo de alojamiento incluido en el
paquete turistico',
`TipoTransporte` VARCHAR(100) NOT NULL COMMENT 'Si cuenta con transporte y de que
tipo',
`idAgencia` INT NOT NULL COMMENT 'identificador de la agencia que contrata el tour
(llave foranea)',
PRIMARY KEY (`idTour`),
INDEX `fk_PaqueteTuristico_AgenciaTurismo1_idx` (`idAgencia` ASC),
CONSTRAINT `fk_PaqueteTuristico_AgenciaTurismo1`
FOREIGN KEY (`idAgencia`)
REFERENCES `1masQR_Tur`.`AgenciaTurismo` (`idAgencia`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `1masQR_Tur`.`Registro_Turista_Agencia`
-----

```

```

CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`Registro_Turista_Agencia` (
  `idTurista` INT NOT NULL COMMENT 'Identificador unico para cada turista registrado en el
sistema',
  `idAgencia` INT NOT NULL COMMENT 'Identificador unico para cada agencia turística
registrada en el sistema',
  `Registro_Turista_AgenciaCol` VARCHAR(100) NOT NULL COMMENT 'Registro del usuario
por parte de la agencia turística',
  `CodigoRegistro` INT NOT NULL COMMENT 'Codigo generado del contrato entre el turista y
la agencia para realizar el paquete turistico',
  `FechaInicioTour` DATE NOT NULL COMMENT 'Fecha de inicio del paquete turistico',
  `FechaFinTour` DATE NOT NULL COMMENT 'Fecha de finalización del paquete turistico',
  PRIMARY KEY (`idTurista`, `idAgencia`),
  INDEX `fk_Turista_has_AgenciaTurismo_AgenciaTurismo1_idx` (`idAgencia` ASC),
  INDEX `fk_Turista_has_AgenciaTurismo_Turista_idx` (`idTurista` ASC),
  CONSTRAINT `fk_Turista_has_AgenciaTurismo_Turista`
FOREIGN KEY (`idTurista`)
REFERENCES `1masQR_Tur`.`Turista` (`idTurista`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Turista_has_AgenciaTurismo_AgenciaTurismo1`
FOREIGN KEY (`idAgencia`)
REFERENCES `1masQR_Tur`.`AgenciaTurismo` (`idAgencia`)

```

```

    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
-----
-- Table `1masQR_Tur`.`Registro_Turista_Alojamiento`
-----
CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`Registro_Turista_Alojamiento` (
  `idTurista` INT NOT NULL COMMENT 'Identificador unico para cada turista registrado en el
sistema',
  `idAlojamiento` INT NOT NULL COMMENT 'Identificador unico para cada Hotel u otro tipo
de alojamiento registrado en el sistema',
  `Codigo_registro` INT NOT NULL COMMENT 'Código del registro del usuario que va a
utilizar los servicios del hotel',
  `FechaIngreso` DATE NOT NULL COMMENT 'Fecha de ingreso del usuario al Hotel',
  `FechaSalida` DATE NOT NULL COMMENT 'Fecha de salida del usuario del hotel',
  PRIMARY KEY (`idTurista`, `idAlojamiento`),
  INDEX `fk_Turista_has_Alojamiento_Alojamiento1_idx` (`idAlojamiento` ASC),
  INDEX `fk_Turista_has_Alojamiento_Turista1_idx` (`idTurista` ASC),
  CONSTRAINT `fk_Turista_has_Alojamiento_Turista1`
    FOREIGN KEY (`idTurista`)
      REFERENCES `1masQR_Tur`.`Turista` (`idTurista`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fk_Turista_has_Alojamiento_Alojamiento1`
    FOREIGN KEY (`idAlojamiento`)
      REFERENCES `1masQR_Tur`.`Alojamiento` (`idAlojamiento`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;
-----
-- Table `1masQR_Tur`.`Turista_Correo`
-----
CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`Turista_Correo` (
  `idTurista` INT NOT NULL COMMENT 'Identificador unico para cada turista registrado en el
sistema (llave foranea)',
  `id_Correo` VARCHAR(45) NOT NULL COMMENT 'Identificador único de cada correo del
turista o usuario',
  `Correo` VARCHAR(50) NOT NULL COMMENT 'Correo del usuario con el formato adecuado',
  INDEX `fk_Turista_Correo_Turista1_idx` (`idTurista` ASC),
  PRIMARY KEY (`id_Correo`),
  CONSTRAINT `fk_Turista_Correo_Turista1`
    FOREIGN KEY (`idTurista`)
      REFERENCES `1masQR_Tur`.`Turista` (`idTurista`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;
-----

```

-- Table `1masQR_Tur`.`Correo_Alojamiento`

```
-----  
CREATE TABLE IF NOT EXISTS `1masQR_Tur`.`Correo_Alojamiento` (  
  `idAlojamiento` INT NOT NULL COMMENT 'Llave primaria, unica para el Hotel',  
  `idCorreo` INT NOT NULL COMMENT 'Identificador único de cada correo del alojamiento',  
  `CorreoAlojamiento` VARCHAR(50) NOT NULL COMMENT 'Correo del hotel u otro tipo de  
hospedaje con el formato adecuado.',  
  INDEX `fk_AlojamientoCorreo_Alojamiento1_idx` (`idAlojamiento` ASC),  
  PRIMARY KEY (`idCorreo`),  
  CONSTRAINT `fk_AlojamientoCorreo_Alojamiento1`  
    FOREIGN KEY (`idAlojamiento`)  
      REFERENCES `1masQR_Tur`.`Alojamiento` (`idAlojamiento`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

CONCLUSIONES

Por medio del presente Sprint y con el acompañamiento de los instructores de la UTP, se interiorizan los conocimientos acerca de cómo instalar y configurar un sistema gestor de base de datos. Además, aprender cómo crear sus tablas y los tipos de datos convenientes según el tipo de información, por ejemplo, para cadenas de texto varchar(50), para identificadores Int, para fechas DateTime, etc. Además, se aprenden los conceptos más importantes de DDL y DML, y también se explica como alimentar la base de datos y aplicar algunas sentencias SQL sobre las tablas de la base de datos creada.

BIBLIOGRAFIA

- Curso MySQL esencial. Linkedin Learning (2019). Cursado y finalizado en Mayo de 2021. Convenio Sena – Linkedin. <https://www.linkedin.com/learning/mysql-esencial>
- Documentación técnica de SQL Server. (2017). Recuperado de <https://docs.microsoft.com/es-es/sql/sql-server/sql-server-technical-documentation>
- Manual de Referencia MySQL. (2017). Recuperado de <http://dev.mysql.com/doc/refman/5.7/en/>
- Dubois, P. (2009). Mysql Fourth Edition. Madrid: Addison-Wesley.

