|  |
| --- |
| **SPRINT 2: Repositorio de Código - Diseño Base de Datos** |

|  |  |
| --- | --- |
| **Identificación Proyecto** | |
| Nombre Proyecto: | **Consultorio médico Online** |
| Número Equipo: | 1 |
| **Integrantes del equipo** | |
| Rol  (Líder-Desarrollador – Cliente) | Nombre |
| Desarrollador – Scrum Master | Jorge Orlando Bogotá |
| Desarrollador | Cristian Murcia Gomez |
| Desarrollador | William Lopez |
| Desarrollador | Ronal Londoño |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| **Repositorio de Código GitLab o GitHub** |

Como evidencia del repositorio de código, creado con GitLab o GitHub, además de la URL del repositorio, se debe presentar capturas de pantalla donde se visualicen aspectos:

* Creación del proyecto del repositorio.
* Integrantes del equipo invitados.
* Evidencia de la realización de alguna actualización (commit), donde se visualice la actualización y el historial de actualizaciones (Versiones)

|  |
| --- |
| **Diseño de la Base de Datos (Proceso de normalización)** |

Como evidencia del Diseño de la Base de Datos, se debe presentar el proceso de normalización efectuado (Formas Normales).

**Primera forma normal**

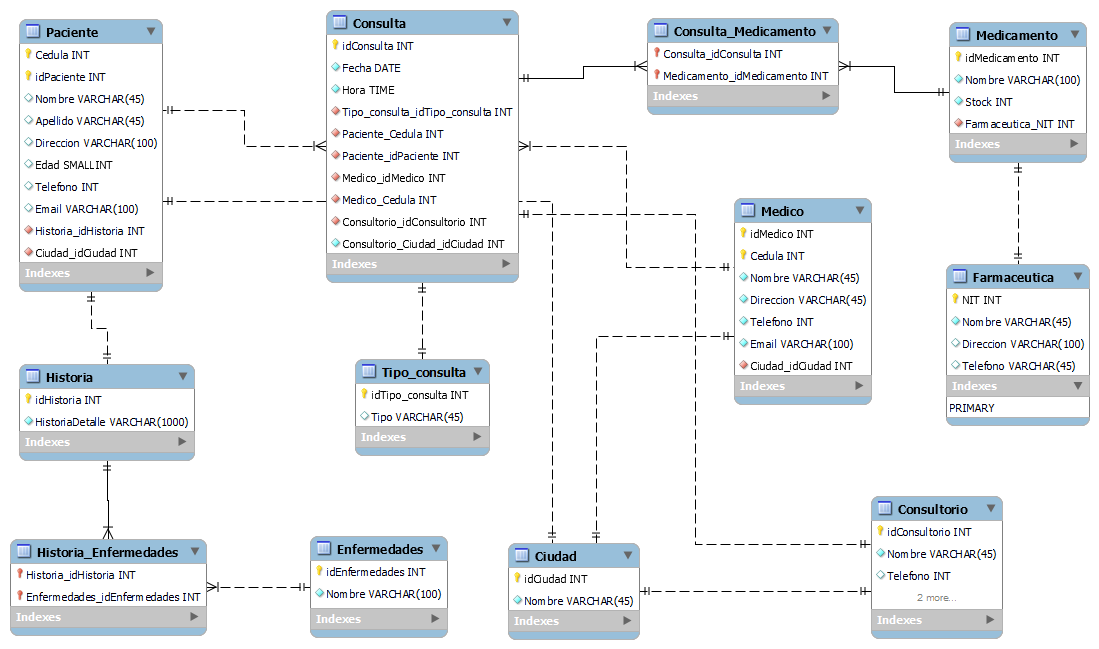
* Elimine los grupos repetidos de las tablas individuales.
* Cree una tabla independiente para cada conjunto de datos relacionados.
* Identifique cada conjunto de datos relacionados con una clave principal.

**Segunda forma normal**

* Cree tablas independientes para conjuntos de valores que se apliquen a varios registros.
* Relacione estas tablas con una clave externa.

**Tercera forma normal**

* Elimine los campos que no dependan de la clave.



|  |
| --- |
| **Esquema de la Base de Datos (Código SQL)** |

Como evidencia del Esquema de la Base de Datos, se debe presentar el código SQL de creación de la base de datos.

-- 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='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

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

-- Schema Consultorio\_Medico

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

-- Base de datos de un consultorio on line para el Ciclo 3 UNAB Mision TIC 2022

--

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

-- Schema Consultorio\_Medico

--

-- Base de datos de un consultorio on line para el Ciclo 3 UNAB Mision TIC 2022

--

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

CREATE SCHEMA IF NOT EXISTS `Consultorio\_Medico` DEFAULT CHARACTER SET utf8 COLLATE utf8\_spanish\_ci ;

USE `Consultorio\_Medico` ;

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

-- Table `Consultorio\_Medico`.`Historia`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Historia` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Historia` (

`idHistoria` INT NOT NULL AUTO\_INCREMENT,

`HistoriaDetalle` VARCHAR(1000) NOT NULL,

PRIMARY KEY (`idHistoria`))

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Ciudad`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Ciudad` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Ciudad` (

`idCiudad` INT NOT NULL,

`Nombre` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idCiudad`))

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Paciente`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Paciente` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Paciente` (

`Cedula` INT NOT NULL,

`idPaciente` INT NOT NULL,

`Nombre` VARCHAR(45) NULL,

`Apellido` VARCHAR(45) NULL,

`Direccion` VARCHAR(100) NULL,

`Edad` SMALLINT NULL,

`Telefono` INT NULL,

`Email` VARCHAR(100) NULL,

`Historia\_idHistoria` INT NOT NULL,

`Ciudad\_idCiudad` INT NOT NULL,

PRIMARY KEY (`Cedula`, `idPaciente`),

INDEX `fk\_Paciente\_Historia1\_idx` (`Historia\_idHistoria` ASC) VISIBLE,

INDEX `fk\_Paciente\_Ciudad1\_idx` (`Ciudad\_idCiudad` ASC) VISIBLE,

CONSTRAINT `fk\_Paciente\_Historia`

FOREIGN KEY (`Historia\_idHistoria`)

REFERENCES `Consultorio\_Medico`.`Historia` (`idHistoria`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Paciente\_Ciudad`

FOREIGN KEY (`Ciudad\_idCiudad`)

REFERENCES `Consultorio\_Medico`.`Ciudad` (`idCiudad`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Tipo\_consulta`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Tipo\_consulta` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Tipo\_consulta` (

`idTipo\_consulta` INT NOT NULL,

`Tipo` VARCHAR(45) NULL,

PRIMARY KEY (`idTipo\_consulta`))

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Medico`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Medico` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Medico` (

`idMedico` INT NOT NULL,

`Cedula` INT NOT NULL,

`Nombre` VARCHAR(45) NOT NULL,

`Direccion` VARCHAR(45) NOT NULL,

`Telefono` INT NOT NULL,

`Email` VARCHAR(100) NOT NULL,

`Ciudad\_idCiudad` INT NOT NULL,

PRIMARY KEY (`idMedico`, `Cedula`),

INDEX `fk\_Medico\_Ciudad1\_idx` (`Ciudad\_idCiudad` ASC) VISIBLE,

CONSTRAINT `fk\_Medico\_Ciudad`

FOREIGN KEY (`Ciudad\_idCiudad`)

REFERENCES `Consultorio\_Medico`.`Ciudad` (`idCiudad`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Consultorio`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Consultorio` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Consultorio` (

`idConsultorio` INT NOT NULL,

`Nombre` VARCHAR(45) NOT NULL,

`Telefono` INT NULL,

`Email` VARCHAR(100) NULL,

`Ciudad\_idCiudad` INT NOT NULL,

PRIMARY KEY (`idConsultorio`),

INDEX `fk\_Consultorio\_Ciudad1\_idx` (`Ciudad\_idCiudad` ASC) VISIBLE,

CONSTRAINT `fk\_Consultorio\_Ciudad`

FOREIGN KEY (`Ciudad\_idCiudad`)

REFERENCES `Consultorio\_Medico`.`Ciudad` (`idCiudad`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Consulta`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Consulta` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Consulta` (

`idConsulta` INT NOT NULL AUTO\_INCREMENT,

`Fecha` DATE NOT NULL,

`Hora` TIME NOT NULL,

`Tipo\_consulta\_idTipo\_consulta` INT NOT NULL,

`Paciente\_Cedula` INT NOT NULL,

`Paciente\_idPaciente` INT NOT NULL,

`Medico\_idMedico` INT NOT NULL,

`Medico\_Cedula` INT NOT NULL,

`Consultorio\_idConsultorio` INT NOT NULL,

`Consultorio\_Ciudad\_idCiudad` INT NOT NULL,

PRIMARY KEY (`idConsulta`),

INDEX `fk\_Consulta\_Paciente1\_idx` (`Paciente\_Cedula` ASC, `Paciente\_idPaciente` ASC) VISIBLE,

INDEX `fk\_Consulta\_Medico1\_idx` (`Medico\_idMedico` ASC, `Medico\_Cedula` ASC) VISIBLE,

INDEX `fk\_Consulta\_Consultorio1\_idx` (`Consultorio\_idConsultorio` ASC) VISIBLE,

INDEX `fk\_Consulta\_Tipo\_consulta\_idx` (`Tipo\_consulta\_idTipo\_consulta` ASC) VISIBLE,

CONSTRAINT `fk\_Consulta\_Tipo\_consulta`

FOREIGN KEY (`Tipo\_consulta\_idTipo\_consulta`)

REFERENCES `Consultorio\_Medico`.`Tipo\_consulta` (`idTipo\_consulta`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Consulta\_Paciente`

FOREIGN KEY (`Paciente\_Cedula` , `Paciente\_idPaciente`)

REFERENCES `Consultorio\_Medico`.`Paciente` (`Cedula` , `idPaciente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Consulta\_Medico`

FOREIGN KEY (`Medico\_idMedico` , `Medico\_Cedula`)

REFERENCES `Consultorio\_Medico`.`Medico` (`idMedico` , `Cedula`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Consulta\_Consultorio`

FOREIGN KEY (`Consultorio\_idConsultorio`)

REFERENCES `Consultorio\_Medico`.`Consultorio` (`idConsultorio`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Farmaceutica`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Farmaceutica` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Farmaceutica` (

`NIT` INT NOT NULL,

`Nombre` VARCHAR(45) NOT NULL,

`Direccion` VARCHAR(100) NULL,

`Telefono` VARCHAR(45) NULL,

PRIMARY KEY (`NIT`))

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Medicamento`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Medicamento` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Medicamento` (

`idMedicamento` INT NOT NULL,

`Nombre` VARCHAR(100) NOT NULL,

`Stock` INT NOT NULL,

`Farmaceutica\_NIT` INT NOT NULL,

PRIMARY KEY (`idMedicamento`),

INDEX `fk\_Medicamento\_Farmaceutica1\_idx` (`Farmaceutica\_NIT` ASC) VISIBLE,

CONSTRAINT `fk\_Medicamento\_Farmaceutica`

FOREIGN KEY (`Farmaceutica\_NIT`)

REFERENCES `Consultorio\_Medico`.`Farmaceutica` (`NIT`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Enfermedades`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Enfermedades` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Enfermedades` (

`idEnfermedades` INT NOT NULL,

`Nombre` VARCHAR(100) NOT NULL,

PRIMARY KEY (`idEnfermedades`))

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Historia\_Enfermedades`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Historia\_Enfermedades` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Historia\_Enfermedades` (

`Historia\_idHistoria` INT NOT NULL,

`Enfermedades\_idEnfermedades` INT NOT NULL,

PRIMARY KEY (`Historia\_idHistoria`, `Enfermedades\_idEnfermedades`),

INDEX `fk\_Historia\_has\_Enfermedades\_Enfermedades1\_idx` (`Enfermedades\_idEnfermedades` ASC) VISIBLE,

INDEX `fk\_Historia\_has\_Enfermedades\_Historia1\_idx` (`Historia\_idHistoria` ASC) VISIBLE,

CONSTRAINT `fk\_Historia\_has\_Enfermedades\_Historia`

FOREIGN KEY (`Historia\_idHistoria`)

REFERENCES `Consultorio\_Medico`.`Historia` (`idHistoria`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Historia\_has\_Enfermedades\_Enfermedades`

FOREIGN KEY (`Enfermedades\_idEnfermedades`)

REFERENCES `Consultorio\_Medico`.`Enfermedades` (`idEnfermedades`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Consultorio\_Medico`.`Consulta\_Medicamento`

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

DROP TABLE IF EXISTS `Consultorio\_Medico`.`Consulta\_Medicamento` ;

CREATE TABLE IF NOT EXISTS `Consultorio\_Medico`.`Consulta\_Medicamento` (

`Consulta\_idConsulta` INT NOT NULL,

`Medicamento\_idMedicamento` INT NOT NULL,

PRIMARY KEY (`Consulta\_idConsulta`, `Medicamento\_idMedicamento`),

INDEX `fk\_Consulta\_has\_Medicamento\_Medicamento1\_idx` (`Medicamento\_idMedicamento` ASC) VISIBLE,

INDEX `fk\_Consulta\_has\_Medicamento\_Consulta1\_idx` (`Consulta\_idConsulta` ASC) VISIBLE,

CONSTRAINT `fk\_Consulta\_has\_Medicamento\_Consulta1`

FOREIGN KEY (`Consulta\_idConsulta`)

REFERENCES `Consultorio\_Medico`.`Consulta` (`idConsulta`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Consulta\_has\_Medicamento\_Medicamento1`

FOREIGN KEY (`Medicamento\_idMedicamento`)

REFERENCES `Consultorio\_Medico`.`Medicamento` (`idMedicamento`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

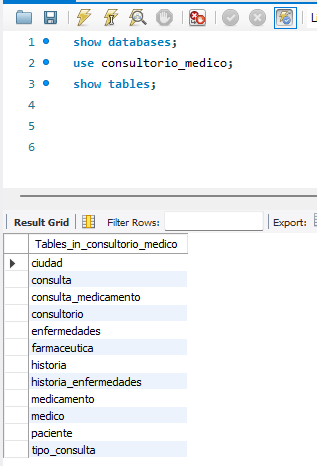
ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

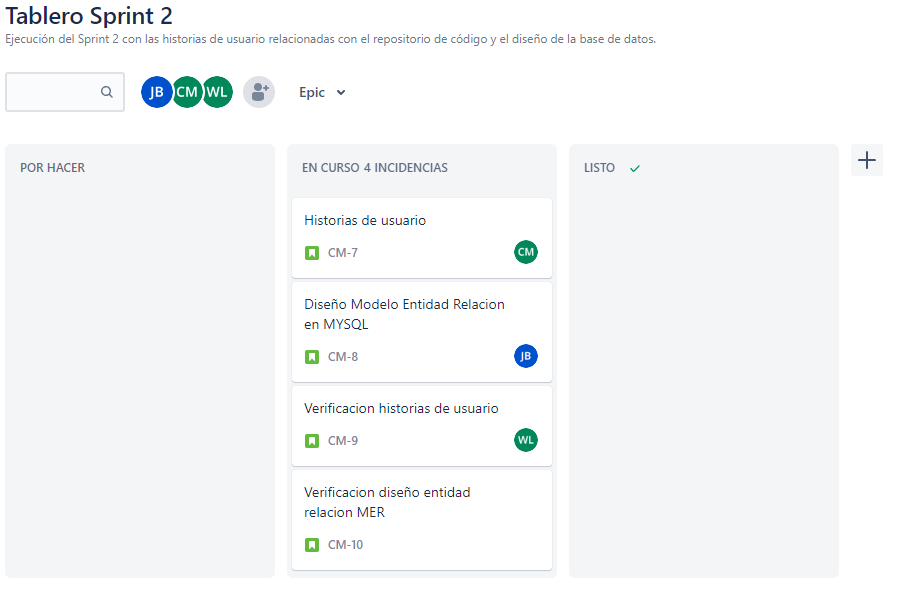
SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**Prueba del cargue de la base de datos:**



|  |
| --- |
| **Evidencia JIRA (Seguimiento del proyecto)** |

Como evidencia del seguimiento del proyecto con la metodología ágil SCRUM, utilizando el software JIRA, se debe presentar capturas de pantalla donde se visualice la ejecución de los Sprint con las historias de usuario relacionadas con el repositorio de código y el diseño de la base de datos.



|  |
| --- |
| **Evidencias de las Reuniones de Equipo** |

Como evidencia de las reuniones que efectúa el equipo del proyecto, presentar capturas de pantalla de las reuniones efectuadas y si lo consideran pertinente algunas actas de las reuniones.

