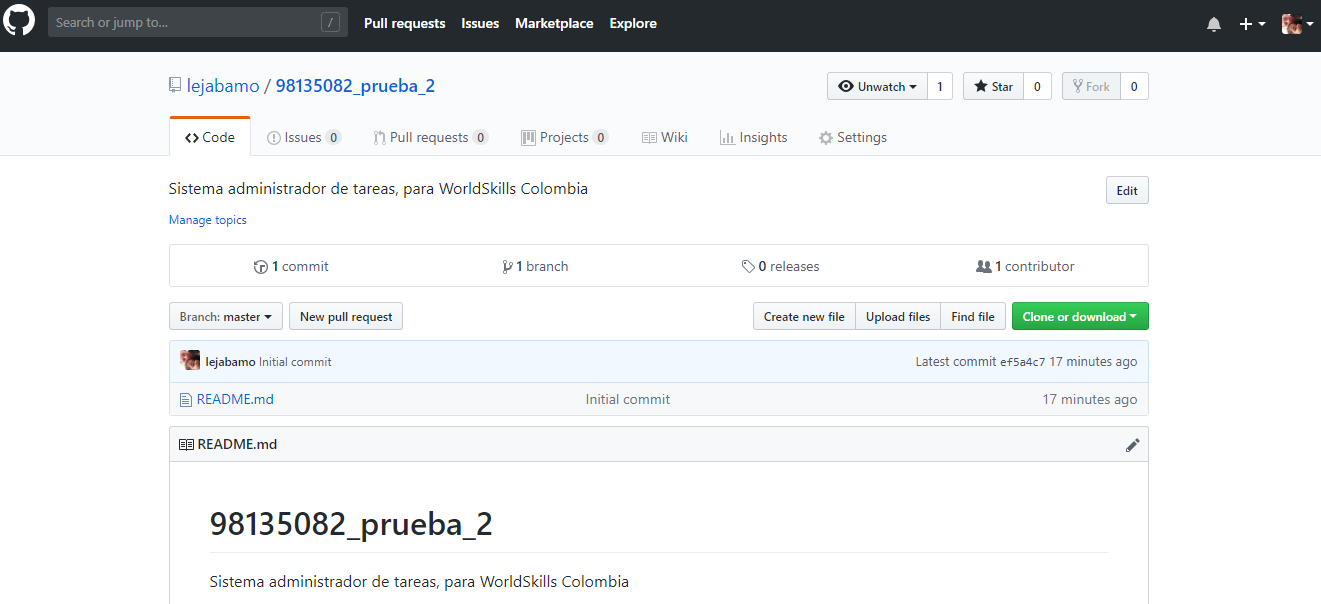
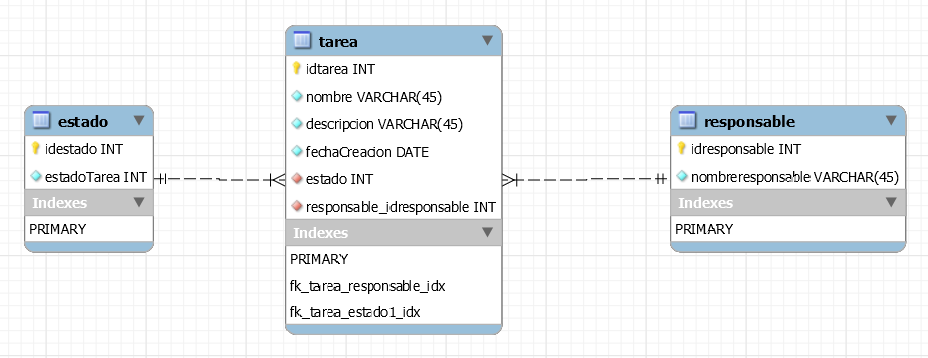
# GESTOR TAREAS WORLD SKILL COLOMBIA

1. El proyecto se configuro en el repositorio de Github como se observa en la siguiente imagen.



1. Diagrama relacional de la base de datos

-Teniendo en cuenta los requerimientos solicitados se presenta el siguiente diagrama.



1. DDL 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='TRADITIONAL,ALLOW\_INVALID\_DATES';

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

-- Schema 98135082\_prueba2

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

-- Modelo Relacional Adminstrador de tareas.

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

-- Schema 98135082\_prueba2

--

-- Modelo Relacional Adminstrador de tareas.

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

CREATE SCHEMA IF NOT EXISTS `98135082\_prueba2` DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci ;

USE `98135082\_prueba2` ;

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

-- Table `98135082\_prueba2`.`responsable`

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

CREATE TABLE IF NOT EXISTS `98135082\_prueba2`.`responsable` (

`idresponsable` INT NOT NULL,

`nombreresponsable` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idresponsable`))

ENGINE = InnoDB;

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

-- Table `98135082\_prueba2`.`estado`

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

CREATE TABLE IF NOT EXISTS `98135082\_prueba2`.`estado` (

`idestado` INT NOT NULL,

`estadoTarea` INT NOT NULL COMMENT 'estadoTarea: \n1. Backlog\n2. En progreso\n3. Suspendido\n4.Cancelado\n5. Terminado',

PRIMARY KEY (`idestado`))

ENGINE = InnoDB;

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

-- Table `98135082\_prueba2`.`tarea`

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

CREATE TABLE IF NOT EXISTS `98135082\_prueba2`.`tarea` (

`idtarea` INT NOT NULL,

`nombre` VARCHAR(45) NOT NULL,

`descripcion` VARCHAR(45) NOT NULL,

`fechaCreacion` DATE NOT NULL,

`estado` INT NOT NULL,

`responsable\_idresponsable` INT NOT NULL,

PRIMARY KEY (`idtarea`),

INDEX `fk\_tarea\_responsable\_idx` (`responsable\_idresponsable` ASC),

INDEX `fk\_tarea\_estado1\_idx` (`estado` ASC),

CONSTRAINT `fk\_tarea\_responsable`

FOREIGN KEY (`responsable\_idresponsable`)

REFERENCES `98135082\_prueba2`.`responsable` (`idresponsable`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_tarea\_estado1`

FOREIGN KEY (`estado`)

REFERENCES `98135082\_prueba2`.`estado` (`idestado`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

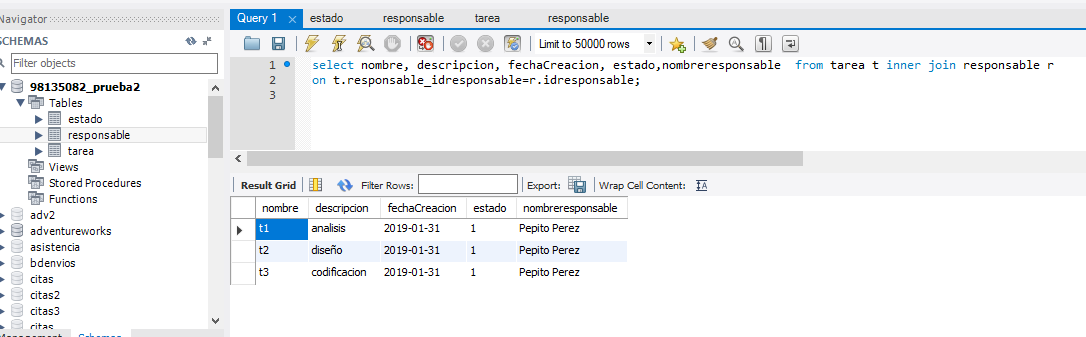
SET SQL\_MODE=@OLD\_SQL\_MODE;

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

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

1. SQL (DML) Trazabilidad tarea incluido responsable

*select nombre, descripcion, fechaCreacion, estado,nombreresponsable from tarea t inner join responsable r on t.responsable\_idresponsable=r.idresponsable;*



1. SQL (DML) Escribir la sentencia SQL que permita obtener el número de tareas de cada responsable.

*select nombreresponsable, count(idtarea) as numeroTareas*

*from tarea t inner join responsable r*

*on t.responsable\_idresponsable=r.idresponsable*

*group by idresponsable;*

