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

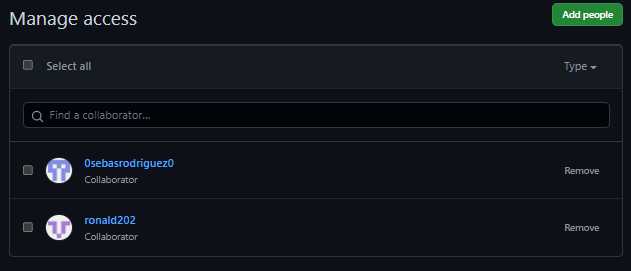
|  |  |
| --- | --- |
| **Identificación Proyecto** | |
| Nombre Proyecto: | Desarrollo de una aplicación web para préstamo de VideoBeams en la universidad de pamplona sede villa del rosario |
| Número Equipo: | Grupo 11 |
| **Integrantes del equipo** | |
| Rol  (Líder-Desarrollador – Cliente) | Nombre |
| Líder, Desarrollador | Luis Alfredo Chuscano Remolina |
| Desarrollador, Diseñador | Ronald Gabriel Ure Carrillo |
| Diseñador, Desarrollador | Jhoan Sebastian Aparicio Rodriguez |
|  |  |
|  |  |
|  |  |
|  |  |

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

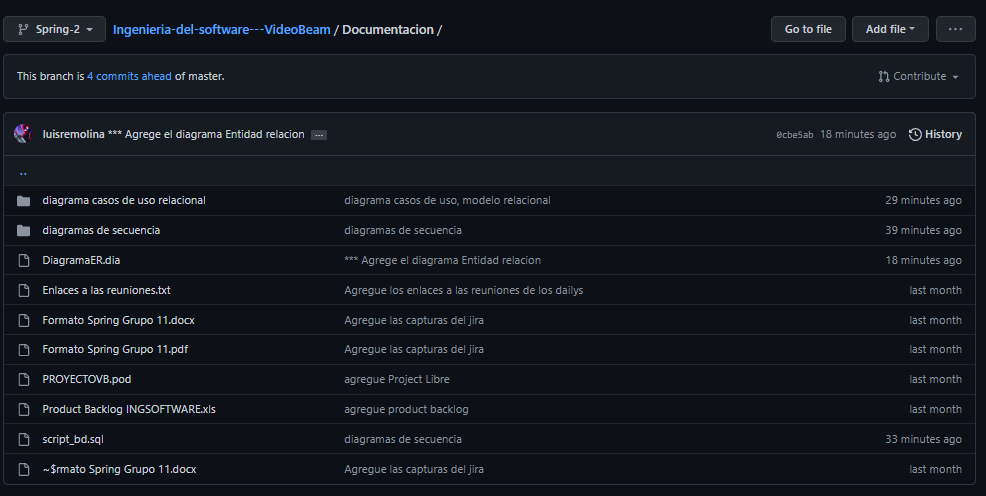
Como evidencia del repositorio de código, creado con GitLab o GitHub, además de la URL del repositorio, se debe presentar capturas de pantalla donde se visualicen aspectos:

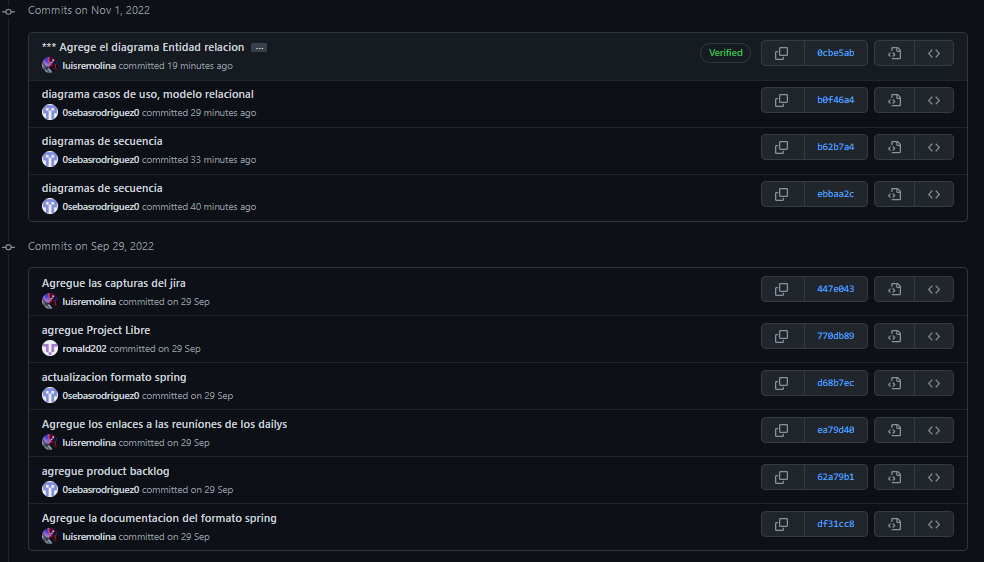
* Creación del proyecto del repositorio.
* Integrantes del equipo invitados.
* Evidencia de la realización de alguna actualización (commit), donde se visualice la actualización y el historial de actualizaciones (Versiones)

<https://github.com/luisremolina/Ingenieria-del-software---VideoBeam/tree/Spring-2>









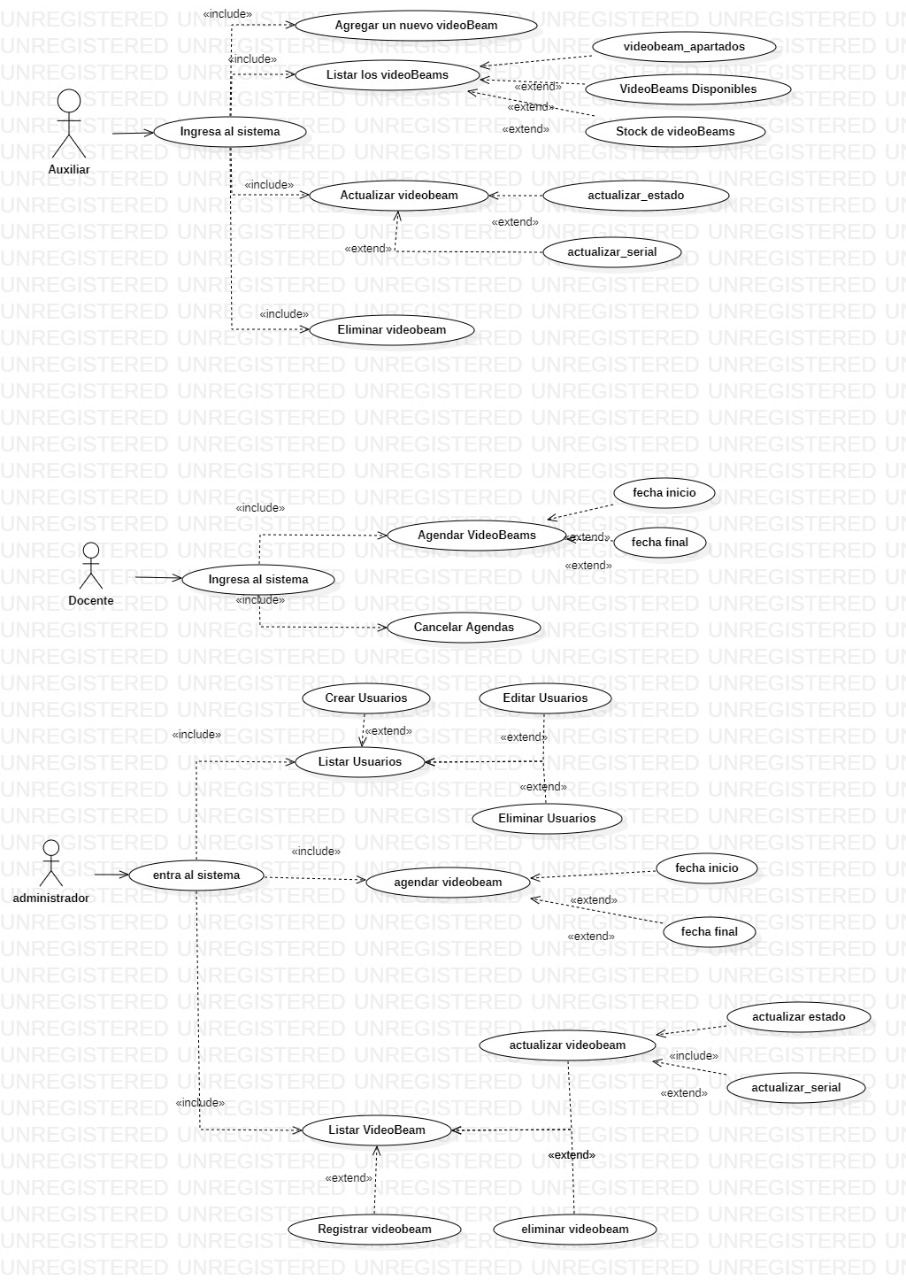
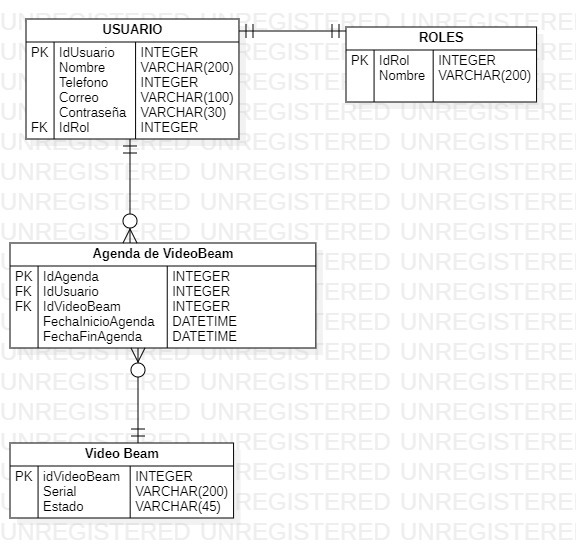
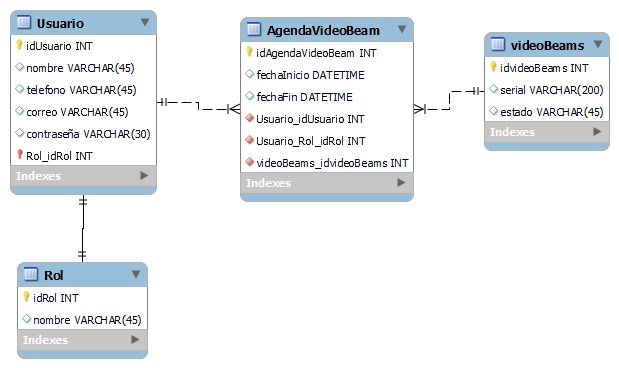
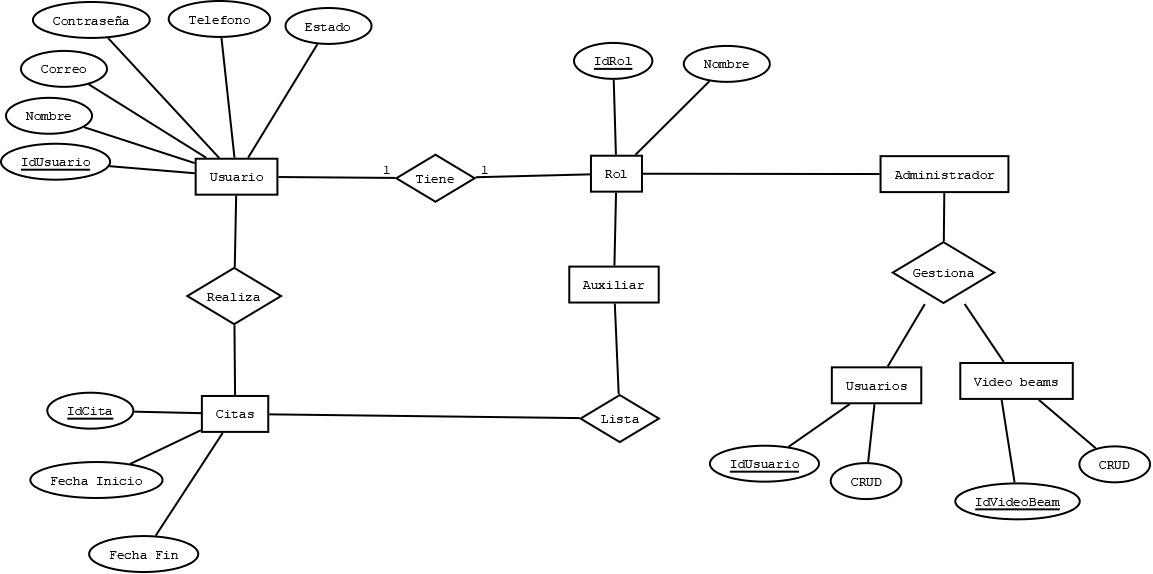
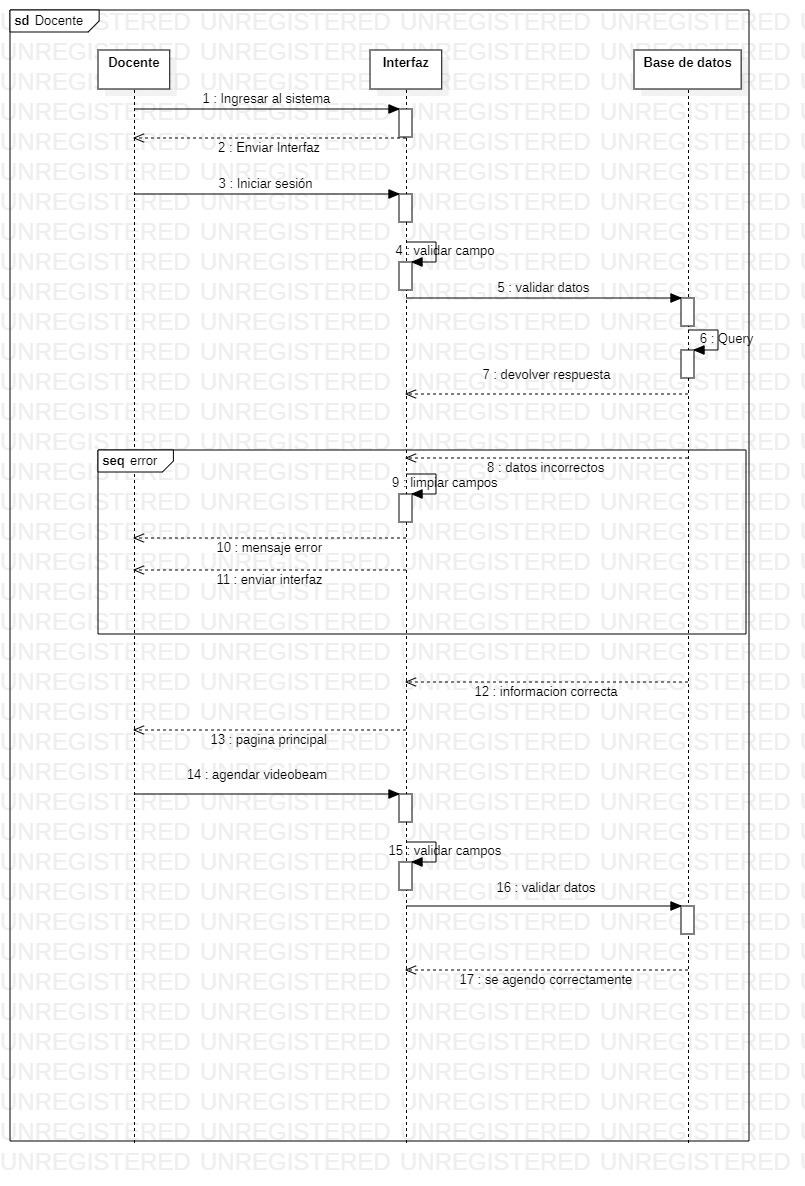
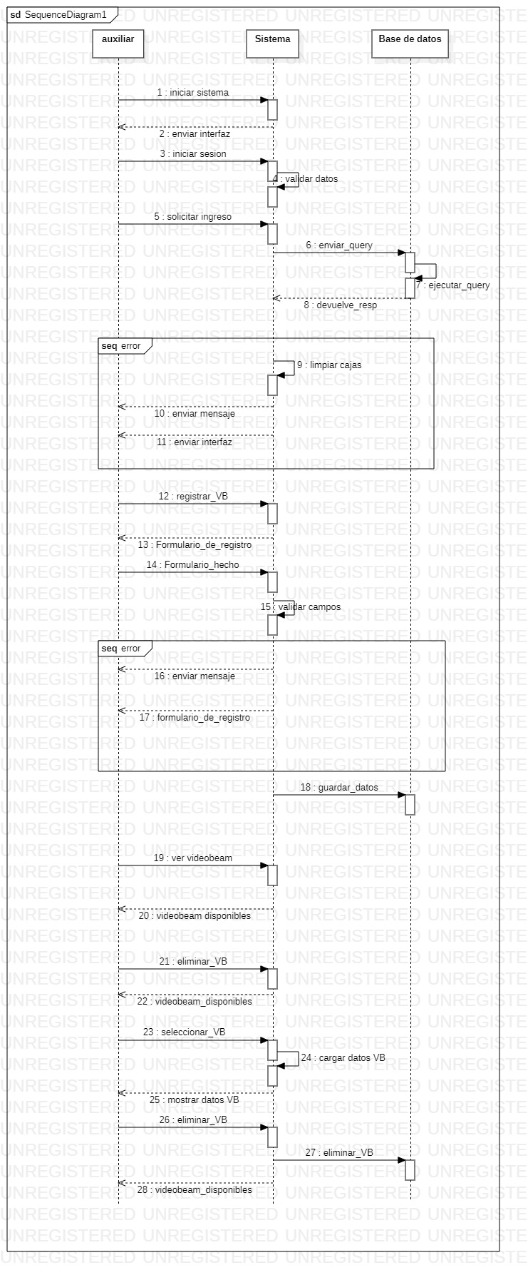
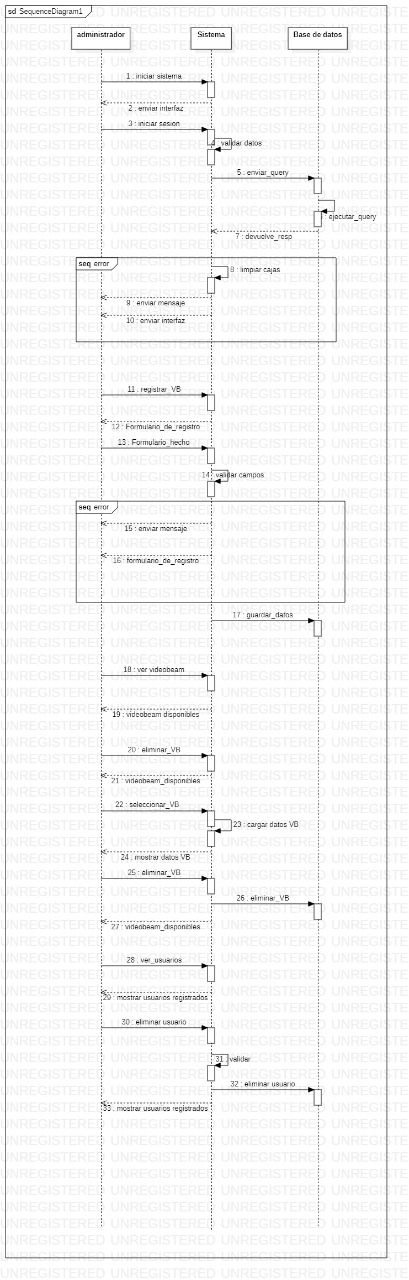
|  |
| --- |
| **Diseño de la Base de Datos (Proceso de normalización)** |

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

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

Se debe presentar:

1. Modelado (Diagrama casos de uso, Diagrama Secuencia, Modelo Entidad relación y Modelo Relacional)



1. Generación del esquema de la Base de Datos Código SQL(Script), mediante el uso del programa Workbench

-- MySQL Script generated by MySQL Workbench

-- Tue Nov 1 22:39:30 2022

-- Model: New Model Version: 1.0

-- 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 mydb

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

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

-- Schema mydb

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

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;

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

-- Table `mydb`.`Rol`

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

CREATE TABLE IF NOT EXISTS `mydb`.`Rol` (

`idRol` INT NOT NULL,

`nombre` VARCHAR(45) NULL,

PRIMARY KEY (`idRol`))

ENGINE = InnoDB;

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

-- Table `mydb`.`Usuario`

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

CREATE TABLE IF NOT EXISTS `mydb`.`Usuario` (

`idUsuario` INT NOT NULL,

`nombre` VARCHAR(45) NULL,

`telefono` VARCHAR(45) NULL,

`correo` VARCHAR(45) NULL,

`contraseña` VARCHAR(30) NULL,

`Rol\_idRol` INT NOT NULL,

PRIMARY KEY (`idUsuario`, `Rol\_idRol`),

INDEX `fk\_Usuario\_Rol\_idx` (`Rol\_idRol` ASC) VISIBLE,

CONSTRAINT `fk\_Usuario\_Rol`

FOREIGN KEY (`Rol\_idRol`)

REFERENCES `mydb`.`Rol` (`idRol`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `mydb`.`videoBeams`

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

CREATE TABLE IF NOT EXISTS `mydb`.`videoBeams` (

`idvideoBeams` INT NOT NULL,

`serial` VARCHAR(200) NULL,

`estado` VARCHAR(45) NULL,

PRIMARY KEY (`idvideoBeams`))

ENGINE = InnoDB;

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

-- Table `mydb`.`AgendaVideoBeam`

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

CREATE TABLE IF NOT EXISTS `mydb`.`AgendaVideoBeam` (

`idAgendaVideoBeam` INT NULL,

`fechaInicio` DATETIME NULL,

`fechaFin` DATETIME NULL,

`Usuario\_idUsuario` INT NOT NULL,

`Usuario\_Rol\_idRol` INT NOT NULL,

`videoBeams\_idvideoBeams` INT NOT NULL,

PRIMARY KEY (`idAgendaVideoBeam`),

INDEX `fk\_AgendaVideoBeam\_Usuario1\_idx` (`Usuario\_idUsuario` ASC, `Usuario\_Rol\_idRol` ASC) VISIBLE,

INDEX `fk\_AgendaVideoBeam\_videoBeams1\_idx` (`videoBeams\_idvideoBeams` ASC) VISIBLE,

CONSTRAINT `fk\_AgendaVideoBeam\_Usuario1`

FOREIGN KEY (`Usuario\_idUsuario` , `Usuario\_Rol\_idRol`)

REFERENCES `mydb`.`Usuario` (`idUsuario` , `Rol\_idRol`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_AgendaVideoBeam\_videoBeams1`

FOREIGN KEY (`videoBeams\_idvideoBeams`)

REFERENCES `mydb`.`videoBeams` (`idvideoBeams`)

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;

|  |
| --- |
| **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.