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

|  |  |
| --- | --- |
| **Identificación Proyecto** | |
| Nombre Proyecto: |  |
| Número Equipo: |  |
| **Integrantes del equipo** | |
| Rol  (Líder-Desarrollador – Cliente) | Nombre |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| **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:

* Evidencia de la realización de alguna actualización (commit), donde se visualice la actualización y el historial de actualizaciones (Versiones)

Version 1

* Creación del proyecto del repositorio.

Graphical user interface, text

Description automatically generated

* Integrantes del equipo invitados.

A screenshot of a computer

Description automatically generated

|  |
| --- |
| **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).

Viernes. Reunirnos para mirar lo del diseño de la base de datos.

**ETAPAS: FORMAS NORMALES**

1. Identificación de tablas y campos
2. Dependencias Funcionales Directas DFD y Transitivas DFT
3. Eliminar las DFT –Realizar relaciones
4. Eliminar Relaciones M:N

A. Identificación de tablas y campos



B. Dependencias Funcionales Directas DFD y Transitivas DFT

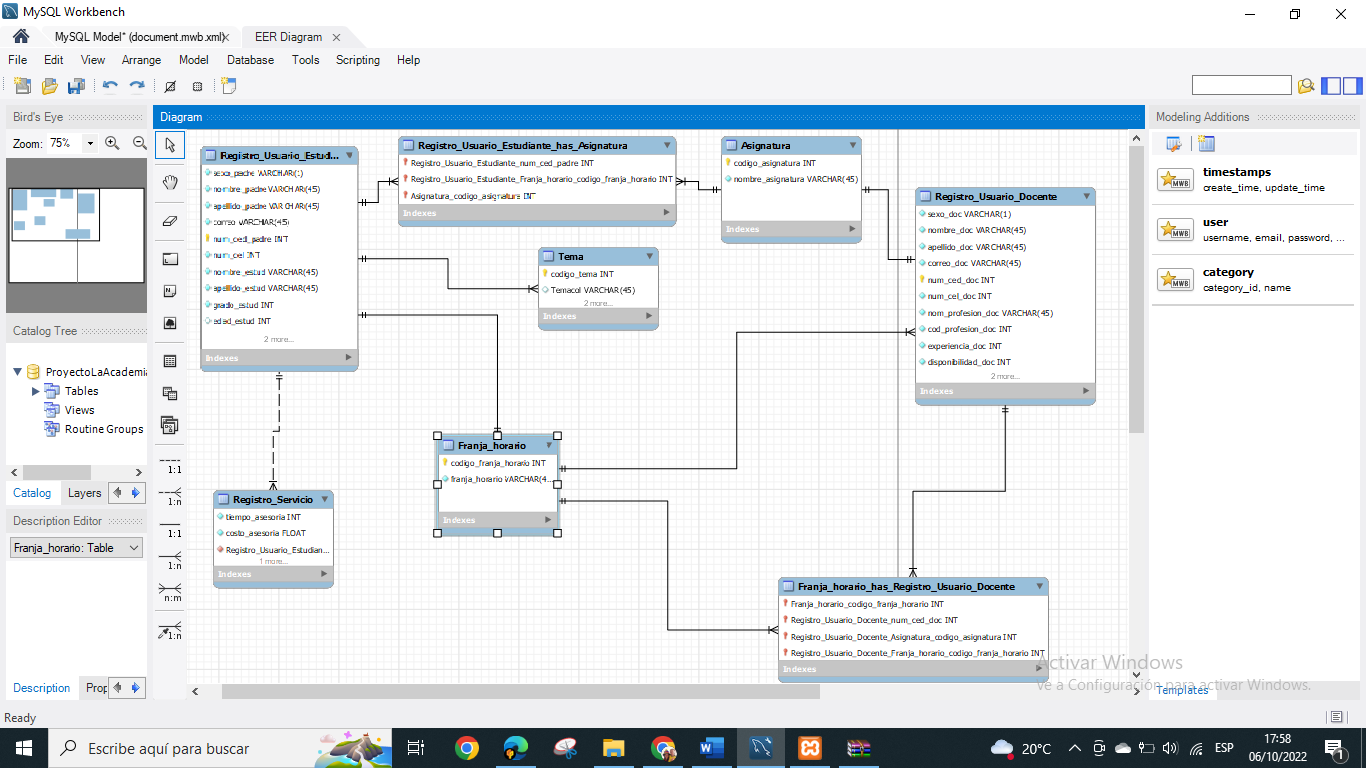


C. Eliminar las DFT –Realizar relaciones

Graphical user interface, application

Description automatically generated

1. Eliminar Relaciones M:N



|  |
| --- |
| **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 ProyectoLaAcademia

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

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

-- Schema ProyectoLaAcademia

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

CREATE SCHEMA IF NOT EXISTS `ProyectoLaAcademia` ;

USE `ProyectoLaAcademia` ;

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

-- Table `ProyectoLaAcademia`.`REGISTO USUARIO ESTUDIANTE`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`REGISTO USUARIO ESTUDIANTE` (

)

ENGINE = InnoDB;

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

-- Table `ProyectoLaAcademia`.`Franja\_horario`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`Franja\_horario` (

`codigo\_franja\_horario` INT NULL,

`franja\_horario` VARCHAR(45) NOT NULL,

PRIMARY KEY (`codigo\_franja\_horario`))

ENGINE = InnoDB;

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

-- Table `ProyectoLaAcademia`.`Registro\_Usuario\_Estudiante`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`Registro\_Usuario\_Estudiante` (

`sexo\_padre` VARCHAR(1) NOT NULL,

`nombre\_padre` VARCHAR(45) NOT NULL,

`apellido\_padre` VARCHAR(45) NOT NULL,

`correo` VARCHAR(45) NOT NULL,

`num\_ced\_padre` INT NULL,

`num\_cel` INT NOT NULL,

`nombre\_estud` VARCHAR(45) NOT NULL,

`apellido\_estud` VARCHAR(45) NOT NULL,

`grado\_estud` INT NOT NULL,

`edad\_estud` INT NULL,

`sexo\_estud` VARCHAR(1) NULL,

`Franja\_horario\_codigo\_franja\_horario` INT NOT NULL,

PRIMARY KEY (`num\_ced\_padre`, `Franja\_horario\_codigo\_franja\_horario`),

INDEX `fk\_Registro\_Usuario\_Estudiante\_Franja\_horario1\_idx` (`Franja\_horario\_codigo\_franja\_horario` ASC) VISIBLE,

CONSTRAINT `fk\_Registro\_Usuario\_Estudiante\_Franja\_horario1`

FOREIGN KEY (`Franja\_horario\_codigo\_franja\_horario`)

REFERENCES `ProyectoLaAcademia`.`Franja\_horario` (`codigo\_franja\_horario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `ProyectoLaAcademia`.`Asignatura`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`Asignatura` (

`codigo\_asignatura` INT NULL,

`nombre\_asignatura` VARCHAR(45) NOT NULL,

PRIMARY KEY (`codigo\_asignatura`))

ENGINE = InnoDB;

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

-- Table `ProyectoLaAcademia`.`Registro\_Usuario\_Docente`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`Registro\_Usuario\_Docente` (

`sexo\_doc` VARCHAR(1) NOT NULL,

`nombre\_doc` VARCHAR(45) NOT NULL,

`apellido\_doc` VARCHAR(45) NOT NULL,

`correo\_doc` VARCHAR(45) NOT NULL,

`num\_ced\_doc` INT NULL,

`num\_cel\_doc` INT NOT NULL,

`nom\_profesion\_doc` VARCHAR(45) NOT NULL,

`cod\_profesion\_doc` INT NOT NULL,

`experiencia\_doc` INT NOT NULL,

`disponibilidad\_doc` INT NOT NULL,

`Asignatura\_codigo\_asignatura` INT NOT NULL,

`Franja\_horario\_codigo\_franja\_horario` INT NOT NULL,

PRIMARY KEY (`num\_ced\_doc`, `Asignatura\_codigo\_asignatura`, `Franja\_horario\_codigo\_franja\_horario`),

INDEX `fk\_Registro\_Usuario\_Docente\_Asignatura1\_idx` (`Asignatura\_codigo\_asignatura` ASC) VISIBLE,

INDEX `fk\_Registro\_Usuario\_Docente\_Franja\_horario1\_idx` (`Franja\_horario\_codigo\_franja\_horario` ASC) VISIBLE,

CONSTRAINT `fk\_Registro\_Usuario\_Docente\_Asignatura1`

FOREIGN KEY (`Asignatura\_codigo\_asignatura`)

REFERENCES `ProyectoLaAcademia`.`Asignatura` (`codigo\_asignatura`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Registro\_Usuario\_Docente\_Franja\_horario1`

FOREIGN KEY (`Franja\_horario\_codigo\_franja\_horario`)

REFERENCES `ProyectoLaAcademia`.`Franja\_horario` (`codigo\_franja\_horario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `ProyectoLaAcademia`.`Registro\_Servicio`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`Registro\_Servicio` (

`tiempo\_asesoria` INT NOT NULL,

`costo\_asesoria` FLOAT NOT NULL,

`Registro\_Usuario\_Estudiante\_num\_ced\_padre` INT NOT NULL,

`Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario` INT NOT NULL,

INDEX `fk\_Registro\_Servicio\_Registro\_Usuario\_Estudiante1\_idx` (`Registro\_Usuario\_Estudiante\_num\_ced\_padre` ASC, `Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario` ASC) VISIBLE,

CONSTRAINT `fk\_Registro\_Servicio\_Registro\_Usuario\_Estudiante1`

FOREIGN KEY (`Registro\_Usuario\_Estudiante\_num\_ced\_padre` , `Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario`)

REFERENCES `ProyectoLaAcademia`.`Registro\_Usuario\_Estudiante` (`num\_ced\_padre` , `Franja\_horario\_codigo\_franja\_horario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `ProyectoLaAcademia`.`Tema`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`Tema` (

`codigo\_tema` INT NULL,

`Temacol` VARCHAR(45) NULL,

`Registro\_Usuario\_Estudiante\_num\_ced\_padre` INT NOT NULL,

`Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario` INT NOT NULL,

PRIMARY KEY (`codigo\_tema`, `Registro\_Usuario\_Estudiante\_num\_ced\_padre`, `Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario`),

INDEX `fk\_Tema\_Registro\_Usuario\_Estudiante1\_idx` (`Registro\_Usuario\_Estudiante\_num\_ced\_padre` ASC, `Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario` ASC) VISIBLE,

CONSTRAINT `fk\_Tema\_Registro\_Usuario\_Estudiante1`

FOREIGN KEY (`Registro\_Usuario\_Estudiante\_num\_ced\_padre` , `Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario`)

REFERENCES `ProyectoLaAcademia`.`Registro\_Usuario\_Estudiante` (`num\_ced\_padre` , `Franja\_horario\_codigo\_franja\_horario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `ProyectoLaAcademia`.`Franja\_horario\_has\_Registro\_Usuario\_Docente`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`Franja\_horario\_has\_Registro\_Usuario\_Docente` (

`Franja\_horario\_codigo\_franja\_horario` INT NOT NULL,

`Registro\_Usuario\_Docente\_num\_ced\_doc` INT NOT NULL,

`Registro\_Usuario\_Docente\_Asignatura\_codigo\_asignatura` INT NOT NULL,

`Registro\_Usuario\_Docente\_Franja\_horario\_codigo\_franja\_horario` INT NOT NULL,

PRIMARY KEY (`Franja\_horario\_codigo\_franja\_horario`, `Registro\_Usuario\_Docente\_num\_ced\_doc`, `Registro\_Usuario\_Docente\_Asignatura\_codigo\_asignatura`, `Registro\_Usuario\_Docente\_Franja\_horario\_codigo\_franja\_horario`),

INDEX `fk\_Franja\_horario\_has\_Registro\_Usuario\_Docente\_Registro\_Usu\_idx` (`Registro\_Usuario\_Docente\_num\_ced\_doc` ASC, `Registro\_Usuario\_Docente\_Asignatura\_codigo\_asignatura` ASC, `Registro\_Usuario\_Docente\_Franja\_horario\_codigo\_franja\_horario` ASC) VISIBLE,

INDEX `fk\_Franja\_horario\_has\_Registro\_Usuario\_Docente\_Franja\_horar\_idx` (`Franja\_horario\_codigo\_franja\_horario` ASC) VISIBLE,

CONSTRAINT `fk\_Franja\_horario\_has\_Registro\_Usuario\_Docente\_Franja\_horario1`

FOREIGN KEY (`Franja\_horario\_codigo\_franja\_horario`)

REFERENCES `ProyectoLaAcademia`.`Franja\_horario` (`codigo\_franja\_horario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Franja\_horario\_has\_Registro\_Usuario\_Docente\_Registro\_Usuar1`

FOREIGN KEY (`Registro\_Usuario\_Docente\_num\_ced\_doc` , `Registro\_Usuario\_Docente\_Asignatura\_codigo\_asignatura` , `Registro\_Usuario\_Docente\_Franja\_horario\_codigo\_franja\_horario`)

REFERENCES `ProyectoLaAcademia`.`Registro\_Usuario\_Docente` (`num\_ced\_doc` , `Asignatura\_codigo\_asignatura` , `Franja\_horario\_codigo\_franja\_horario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `ProyectoLaAcademia`.`Registro\_Usuario\_Estudiante\_has\_Asignatura`

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

CREATE TABLE IF NOT EXISTS `ProyectoLaAcademia`.`Registro\_Usuario\_Estudiante\_has\_Asignatura` (

`Registro\_Usuario\_Estudiante\_num\_ced\_padre` INT NOT NULL,

`Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario` INT NOT NULL,

`Asignatura\_codigo\_asignatura` INT NOT NULL,

PRIMARY KEY (`Registro\_Usuario\_Estudiante\_num\_ced\_padre`, `Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario`, `Asignatura\_codigo\_asignatura`),

INDEX `fk\_Registro\_Usuario\_Estudiante\_has\_Asignatura\_Asignatura1\_idx` (`Asignatura\_codigo\_asignatura` ASC) VISIBLE,

INDEX `fk\_Registro\_Usuario\_Estudiante\_has\_Asignatura\_Registro\_Usua\_idx` (`Registro\_Usuario\_Estudiante\_num\_ced\_padre` ASC, `Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario` ASC) VISIBLE,

CONSTRAINT `fk\_Registro\_Usuario\_Estudiante\_has\_Asignatura\_Registro\_Usuari1`

FOREIGN KEY (`Registro\_Usuario\_Estudiante\_num\_ced\_padre` , `Registro\_Usuario\_Estudiante\_Franja\_horario\_codigo\_franja\_horario`)

REFERENCES `ProyectoLaAcademia`.`Registro\_Usuario\_Estudiante` (`num\_ced\_padre` , `Franja\_horario\_codigo\_franja\_horario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Registro\_Usuario\_Estudiante\_has\_Asignatura\_Asignatura1`

FOREIGN KEY (`Asignatura\_codigo\_asignatura`)

REFERENCES `ProyectoLaAcademia`.`Asignatura` (`codigo\_asignatura`)

ON DELETE NO ACTION

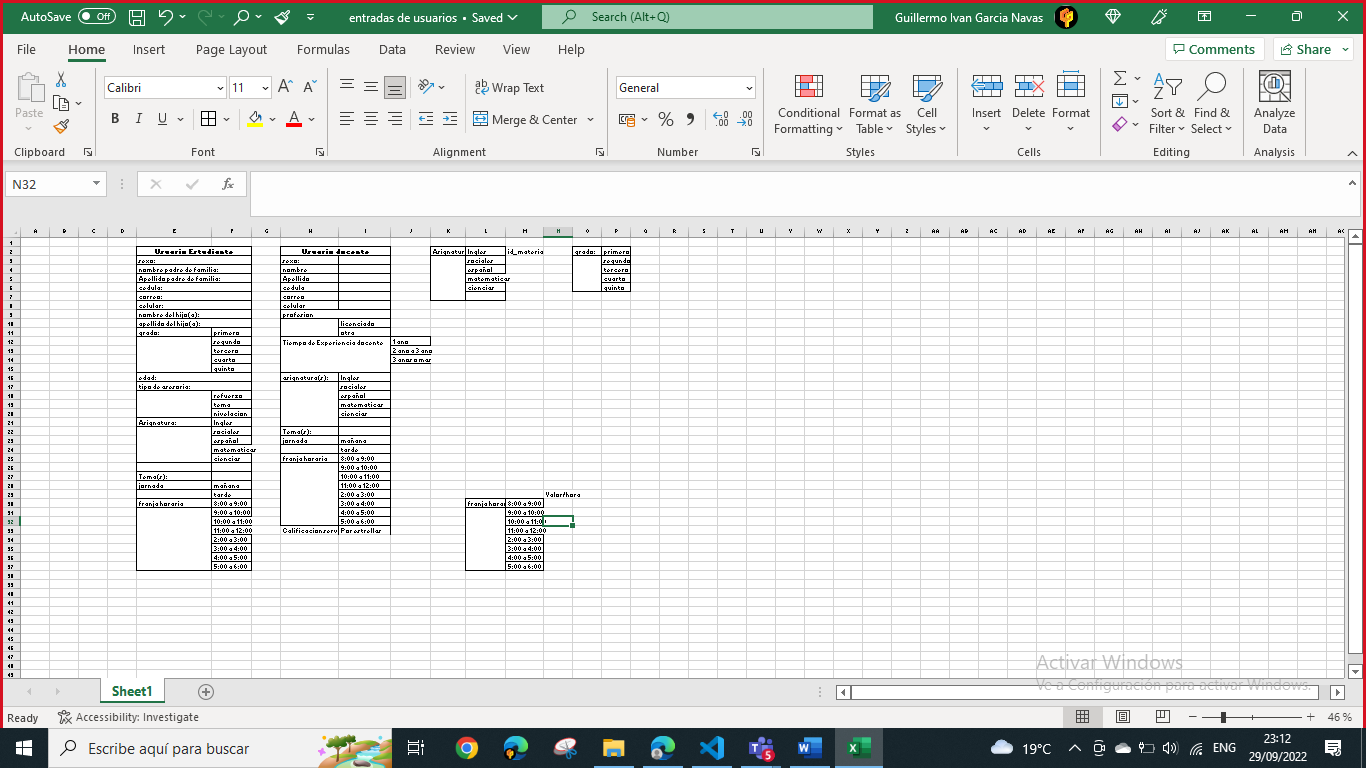
ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

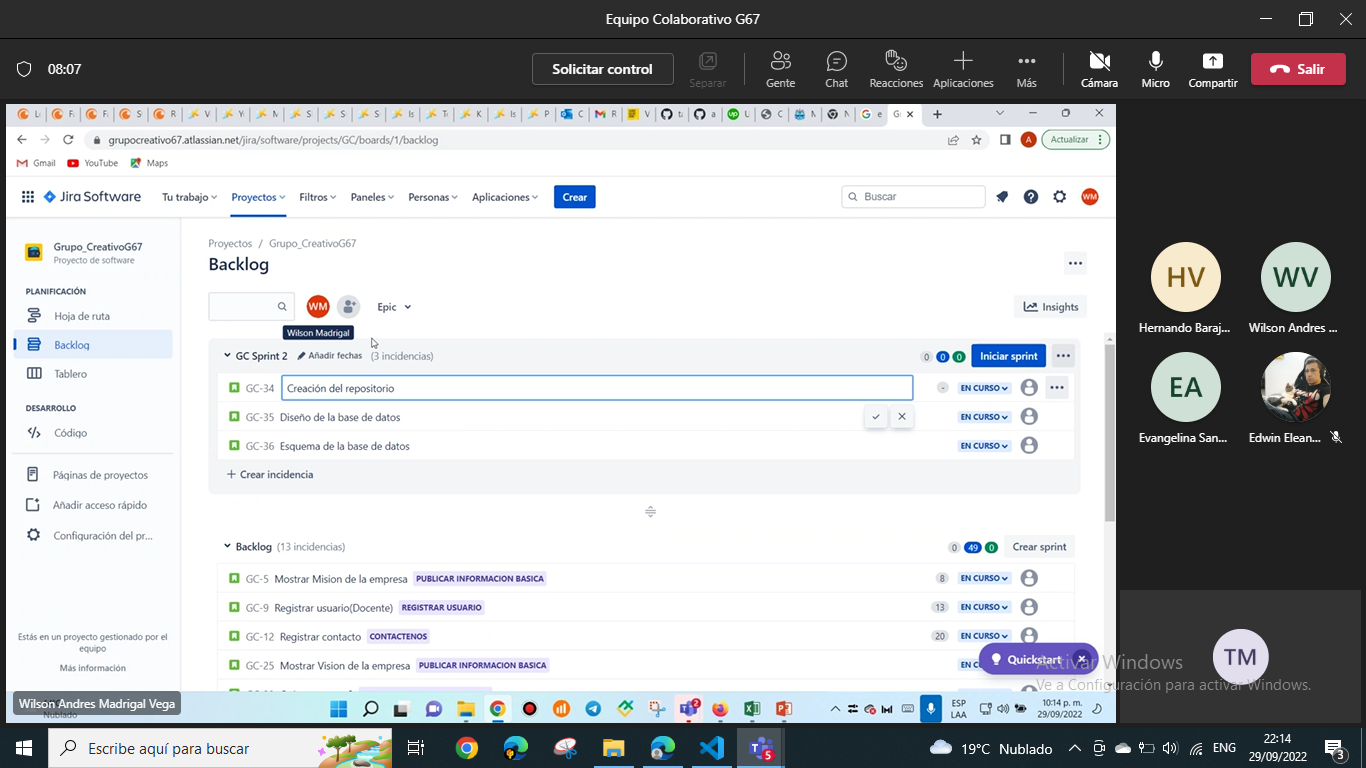
SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

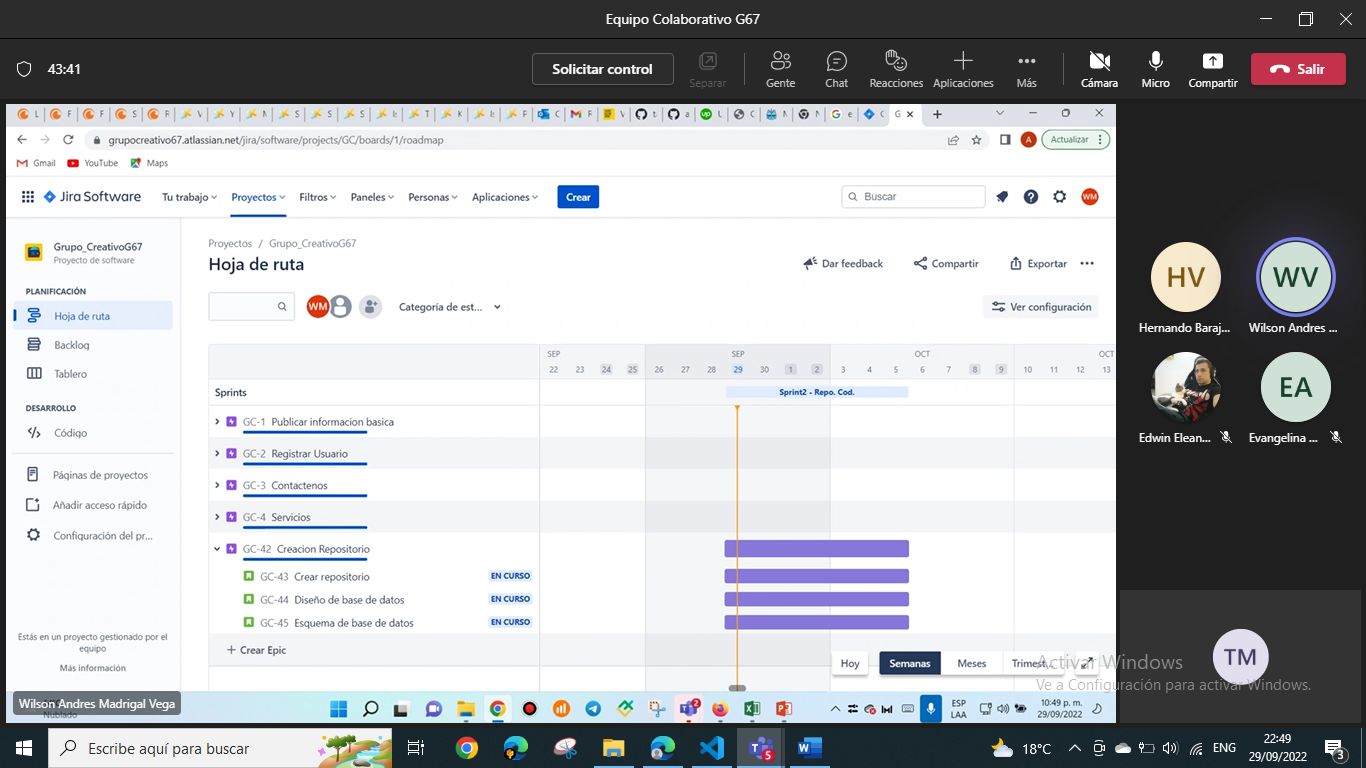
SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

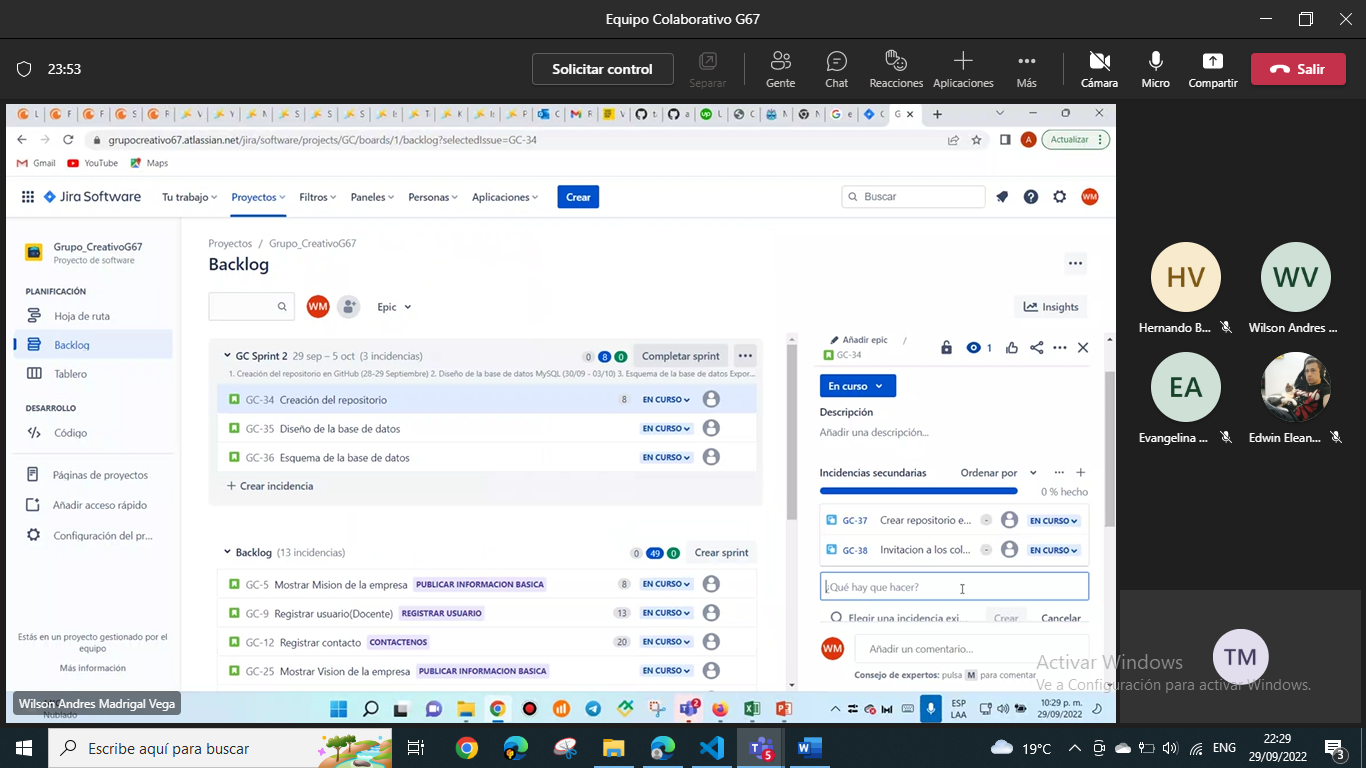


|  |
| --- |
| **Evidencia JIRA (Seguimiento del proyecto)** |

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.

Reunión 27/09/2022

A screenshot of a video game

Description automatically generated

Se realiza la lectura del documento del entregable del Sprint2 para identificar los entregables y definir reuniones y actividades.

Reunión 28/09/2022

Definir los entregables definimos fechas para realizar las actividades. Se deja de tarea revisar las entradas para cada tipo de usuario, para realizar el esquema de la base de datos.

Graphical user interface, application

Description automatically generated

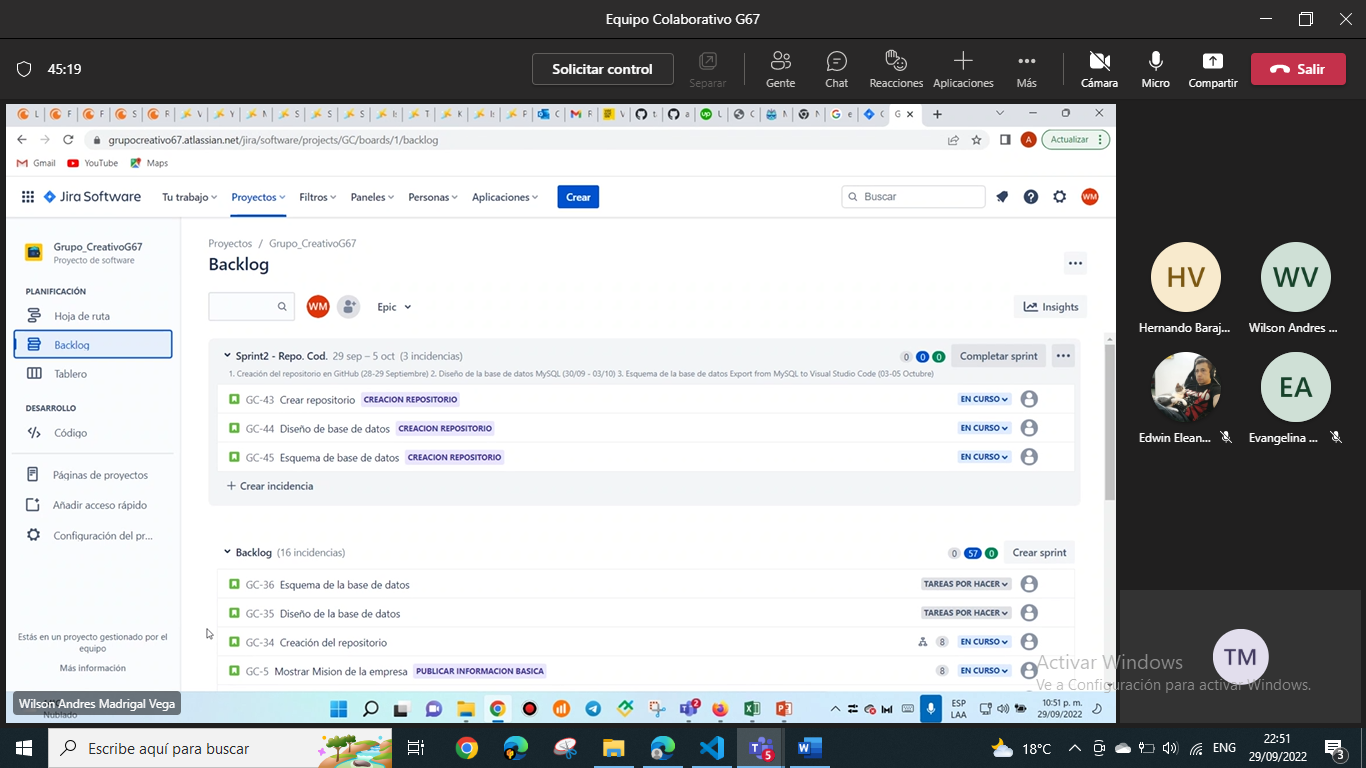
20/09/2022

Actualización de Jira: se monta el Sprint2

A screenshot of a computer

Description automatically generated with medium confidence



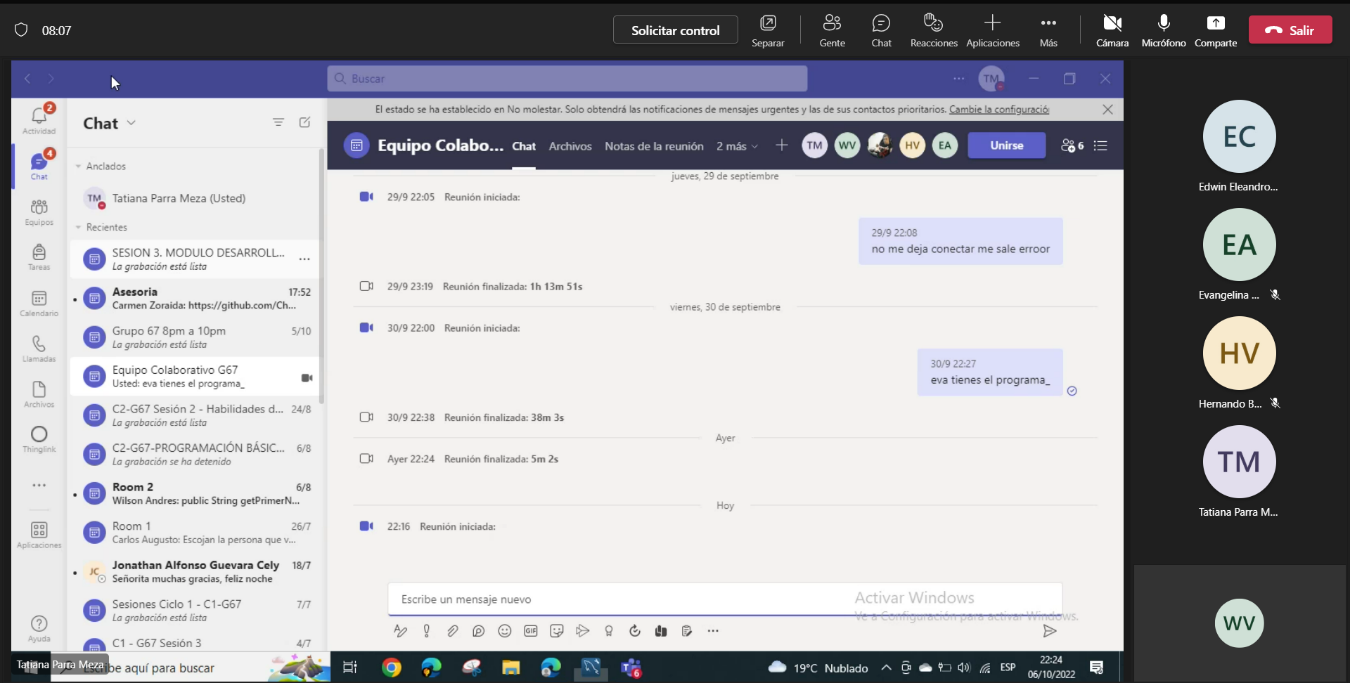


30/09/2022

Reúne el equipo para revisar el diseño de la base de datos. Fred queda como responsable de realizar el borrador de la normalización con la entrada de usuario docente y estudiante.



06/10/2022



A screenshot of a computer

Description automatically generated

Graphical user interface, application

Description automatically generated