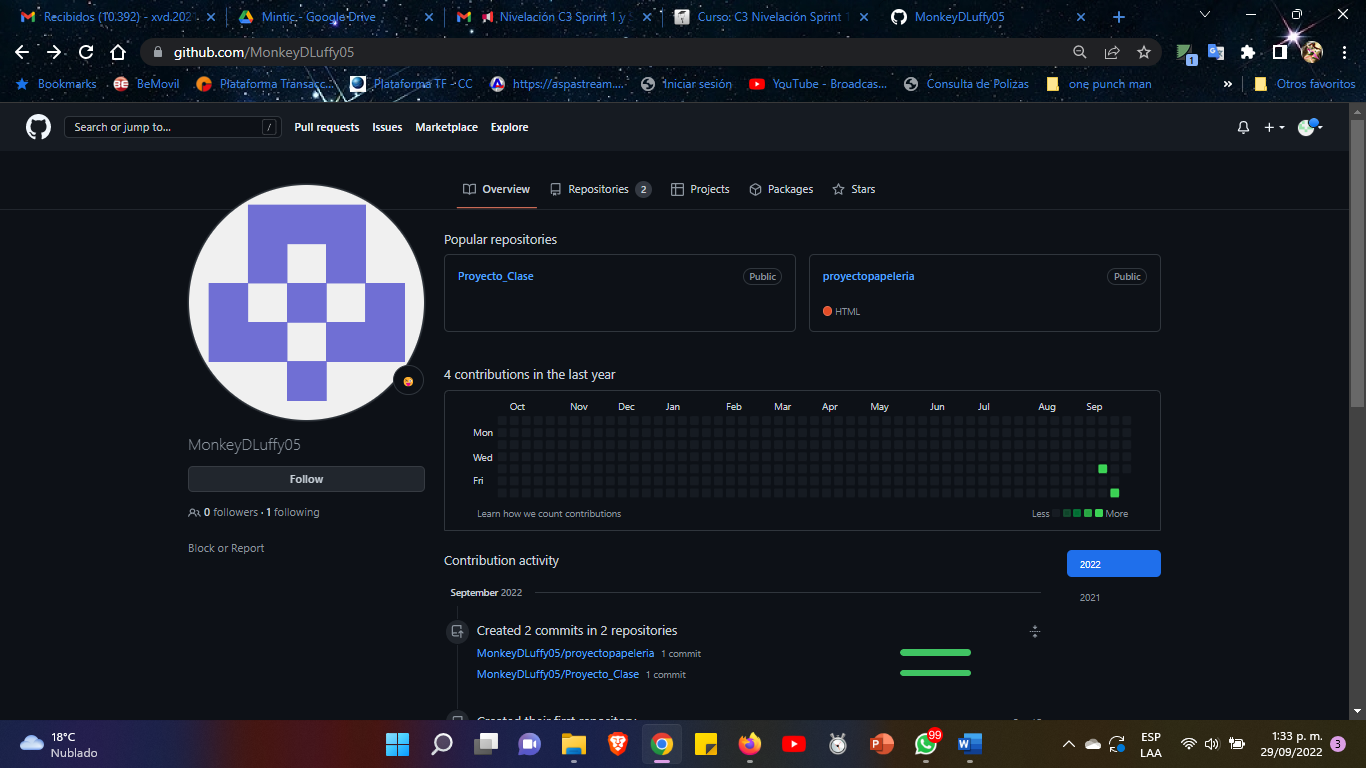
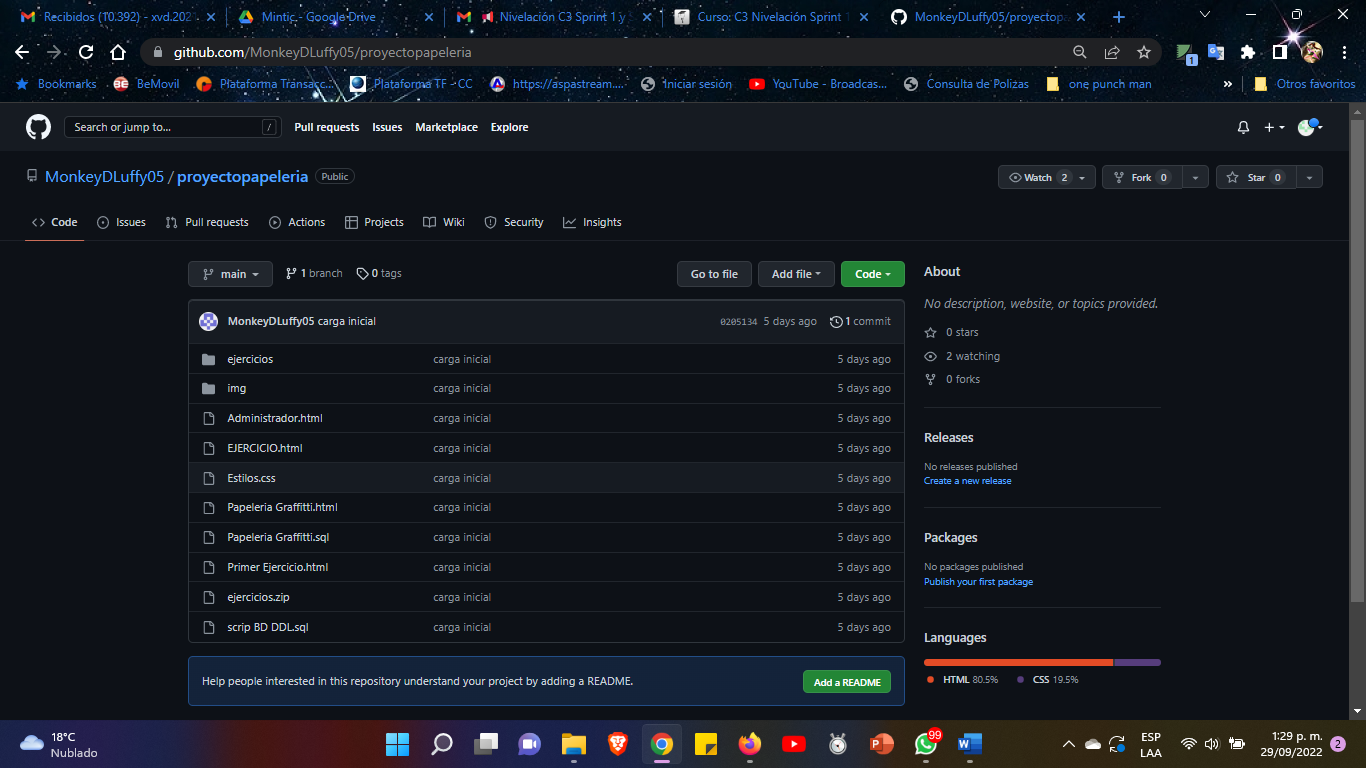
| **SPRINT 2: Repositorio de Código - Diseño Base de Datos** |
| --- |

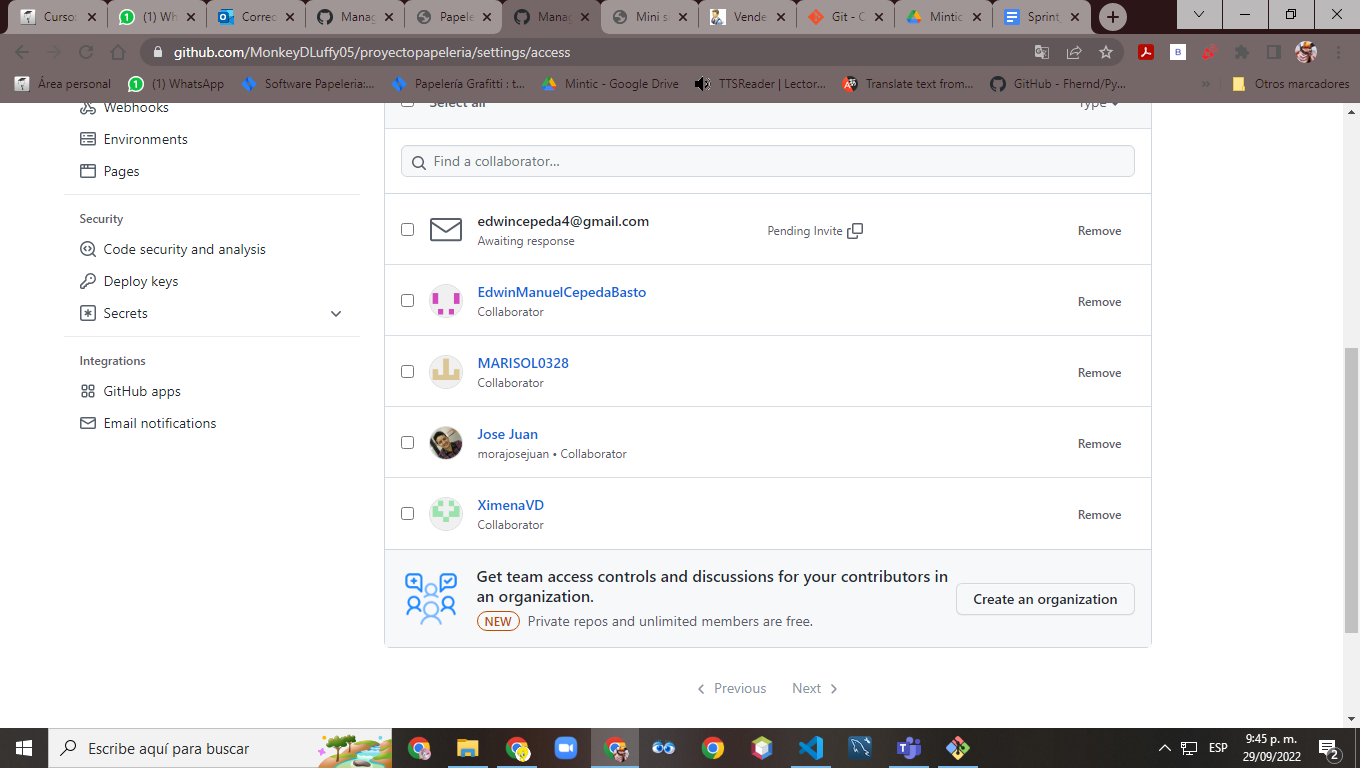
| **Identificación Proyecto** | |
| --- | --- |
| Nombre Proyecto: | Papelería Grafitti |
| Número Equipo: | Equipo 1(uno) |
| **Integrantes del equipo** | |
| Rol  (Líder-Desarrollador – Cliente) | Nombre |
| Líder | Jose Juan Mora |
| Desarrollador-Analista | Ximena Velasquez |
| Desarrollador-Analista | Edwin Manuel Cepeda Basto |
| Desarrollador-Analista | Oscar Mendoza |
| Cliente | Marisol Medina |
|  |  |
|  |  |

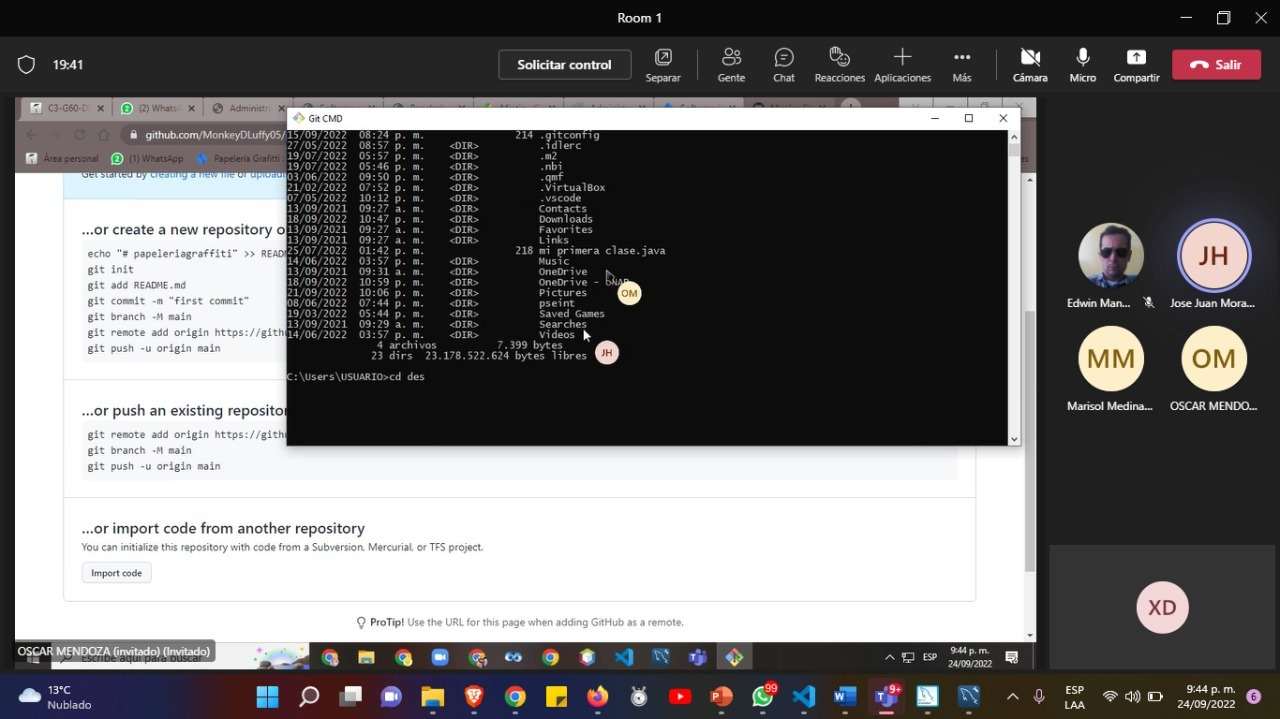
| **Repositorio de Código GitLab o GitHub** |
| --- |

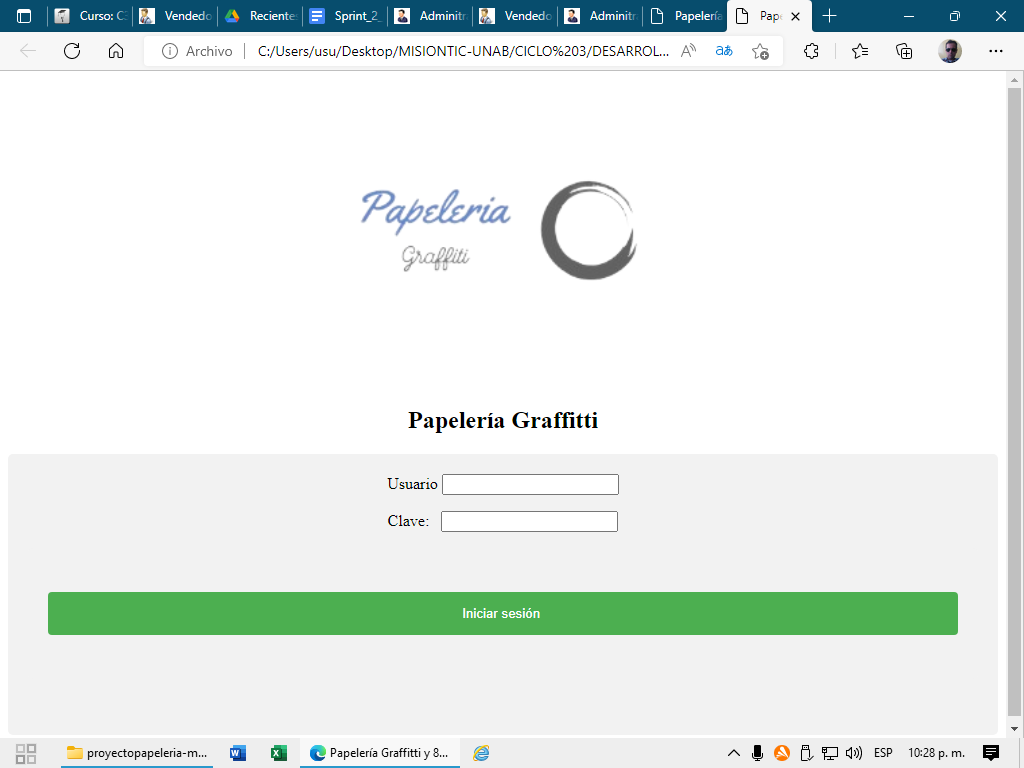
Como evidencia del repositorio de código, creado con GitLab o GitHub, además de la URL del repositorio, se debe presentar capturas de pantalla donde se visualicen aspectos:

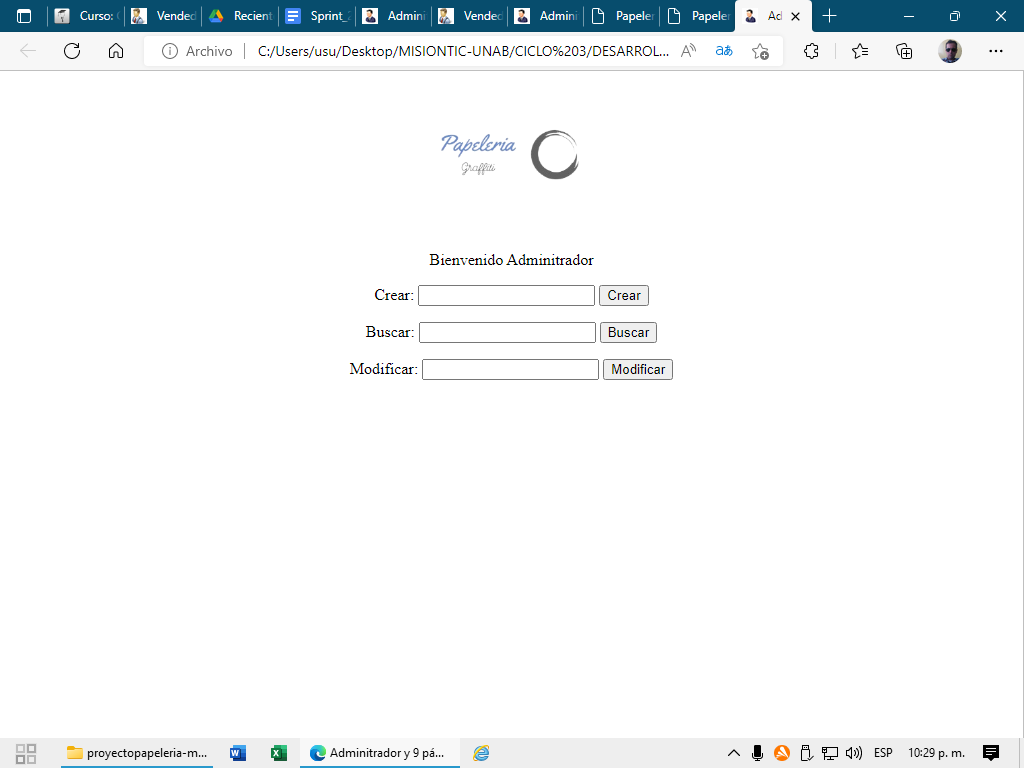
* Creación del proyecto del repositorio.
* Integrantes del equipo invitados.
* Evidencia de la realización de alguna actualización (commit), donde se visualice la actualización y el historial de actualizaciones (Versiones)

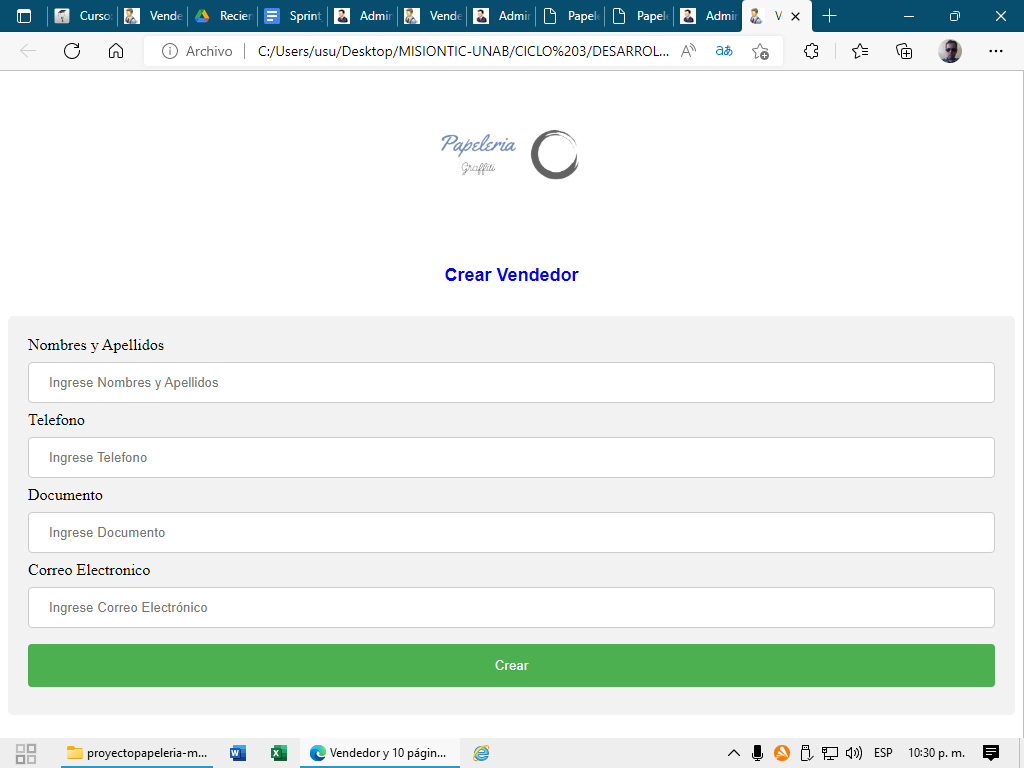
**https://github.com/MonkeyDLuffy05/proyectopapeleria**





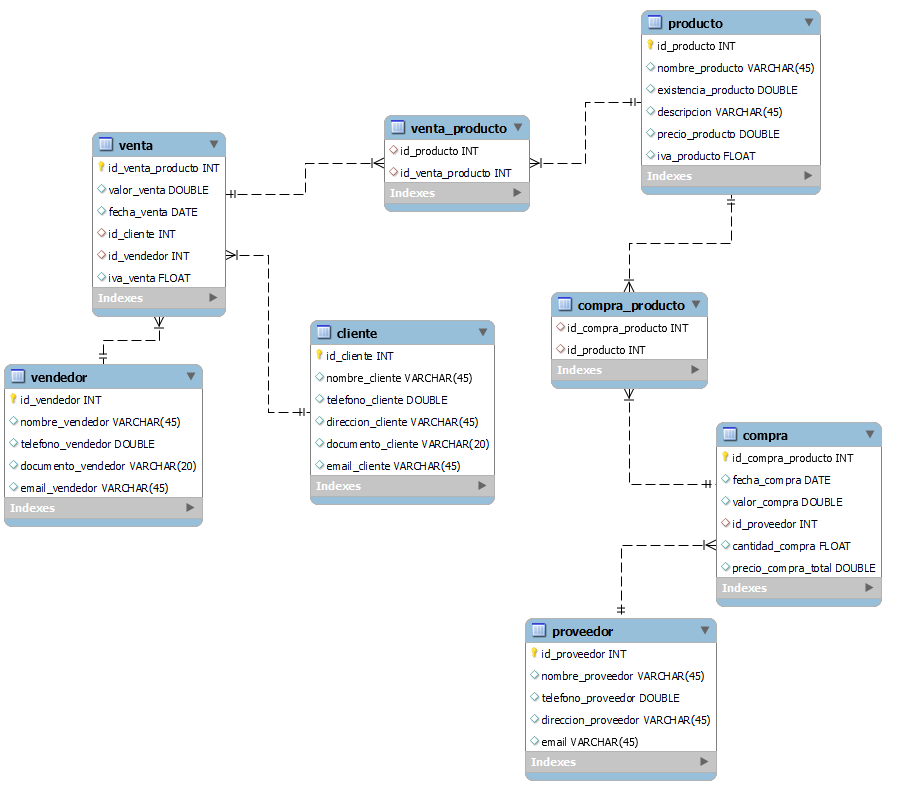






| **Diseño de la Base de Datos (Proceso de normalización)** |
| --- |

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)** |
| --- |

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

create database grafitti\_papeleria;

use grafitti\_papeleria;

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

-- Table `grafitti\_papeleria`.`producto`

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

CREATE TABLE IF NOT EXISTS `grafitti\_papeleria`.`producto` (

`id\_producto` INT NOT NULL,

`nombre\_producto` VARCHAR(45) NULL,

`existencia\_producto` DOUBLE NULL,

`descripcion` VARCHAR(45) NULL,

`precio\_producto` DOUBLE NULL,

`iva\_producto` FLOAT NULL,

PRIMARY KEY (`id\_producto`));

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

-- Table `grafitti\_papeleria`.`vendedor`

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

CREATE TABLE IF NOT EXISTS `grafitti\_papeleria`.`vendedor` (

`id\_vendedor` INT NOT NULL,

`nombre\_vendedor` VARCHAR(45) NULL,

`telefono\_vendedor` DOUBLE NULL,

`documento\_vendedor` VARCHAR(20) NULL,

`email\_vendedor` VARCHAR(45) NULL,

PRIMARY KEY (`id\_vendedor`));

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

-- Table `grafitti\_papeleria`.`cliente`

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

CREATE TABLE IF NOT EXISTS `grafitti\_papeleria`.`cliente` (

`id\_cliente` INT NOT NULL,

`nombre\_cliente` VARCHAR(45) NULL,

`telefono\_cliente` DOUBLE NULL,

`direccion\_cliente` VARCHAR(45) NULL,

`documento\_cliente` VARCHAR(20) NULL,

`email\_cliente` VARCHAR(45) NULL,

PRIMARY KEY (`id\_cliente`));

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

-- Table `grafitti\_papeleria`.`venta`

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

CREATE TABLE IF NOT EXISTS `grafitti\_papeleria`.`venta` (

`id\_venta\_producto` INT NOT NULL,

`valor\_venta` DOUBLE NULL,

`fecha\_venta` DATE NULL,

`id\_cliente` INT NULL,

`id\_vendedor` INT NULL,

`iva\_venta` FLOAT NULL,

PRIMARY KEY (`id\_venta\_producto`),

CONSTRAINT `id\_venta\_vendedor\_fk`

FOREIGN KEY (`id\_vendedor`)

REFERENCES `grafitti\_papeleria`.`vendedor` (`id\_vendedor`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `id\_venta\_cliente\_fk`

FOREIGN KEY (`id\_cliente`)

REFERENCES `grafitti\_papeleria`.`cliente` (`id\_cliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

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

-- Table `grafitti\_papeleria`.`venta\_producto`

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

CREATE TABLE IF NOT EXISTS `grafitti\_papeleria`.`venta\_producto` (

`id\_producto` INT NULL,

`id\_venta\_producto` INT NULL,

CONSTRAINT `id-venta\_producto\_producto\_fk`

FOREIGN KEY (`id\_producto`)

REFERENCES `grafitti\_papeleria`.`producto` (`id\_producto`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `id\_venta\_producto\_venta\_fk`

FOREIGN KEY (`id\_venta\_producto`)

REFERENCES `grafitti\_papeleria`.`venta` (`id\_venta\_producto`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

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

-- Table `grafitti\_papeleria`.`proveedor`

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

CREATE TABLE IF NOT EXISTS `grafitti\_papeleria`.`proveedor` (

`id\_proveedor` INT NOT NULL,

`nombre\_proveedor` VARCHAR(45) NULL,

`telefono\_proveedor` DOUBLE NULL,

`direccion\_proveedor` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

PRIMARY KEY (`id\_proveedor`));

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

-- Table `grafitti\_papeleria`.`compra`

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

CREATE TABLE IF NOT EXISTS `grafitti\_papeleria`.`compra` (

`id\_compra\_producto` INT NOT NULL,

`fecha\_compra` DATE NULL,

`valor\_compra` DOUBLE NULL,

`id\_proveedor` INT NULL,

`cantidad\_compra` FLOAT NULL,

`precio\_compra\_total` DOUBLE NULL,

PRIMARY KEY (`id\_compra\_producto`),

CONSTRAINT `id\_compra\_producto\_proveedor\_fk`

FOREIGN KEY (`id\_proveedor`)

REFERENCES `grafitti\_papeleria`.`proveedor` (`id\_proveedor`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

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

-- Table `grafitti\_papeleria`.`compra\_producto`

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

CREATE TABLE IF NOT EXISTS `grafitti\_papeleria`.`compra\_producto` (

`id\_compra\_producto` INT NULL,

`id\_producto` INT NULL,

CONSTRAINT `id-compra\_producto\_producto\_fk`

FOREIGN KEY (`id\_producto`)

REFERENCES `grafitti\_papeleria`.`producto` (`id\_producto`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `id\_compra\_producto\_compra\_fk`

FOREIGN KEY (`id\_compra\_producto`)

REFERENCES `grafitti\_papeleria`.`compra` (`id\_compra\_producto`)

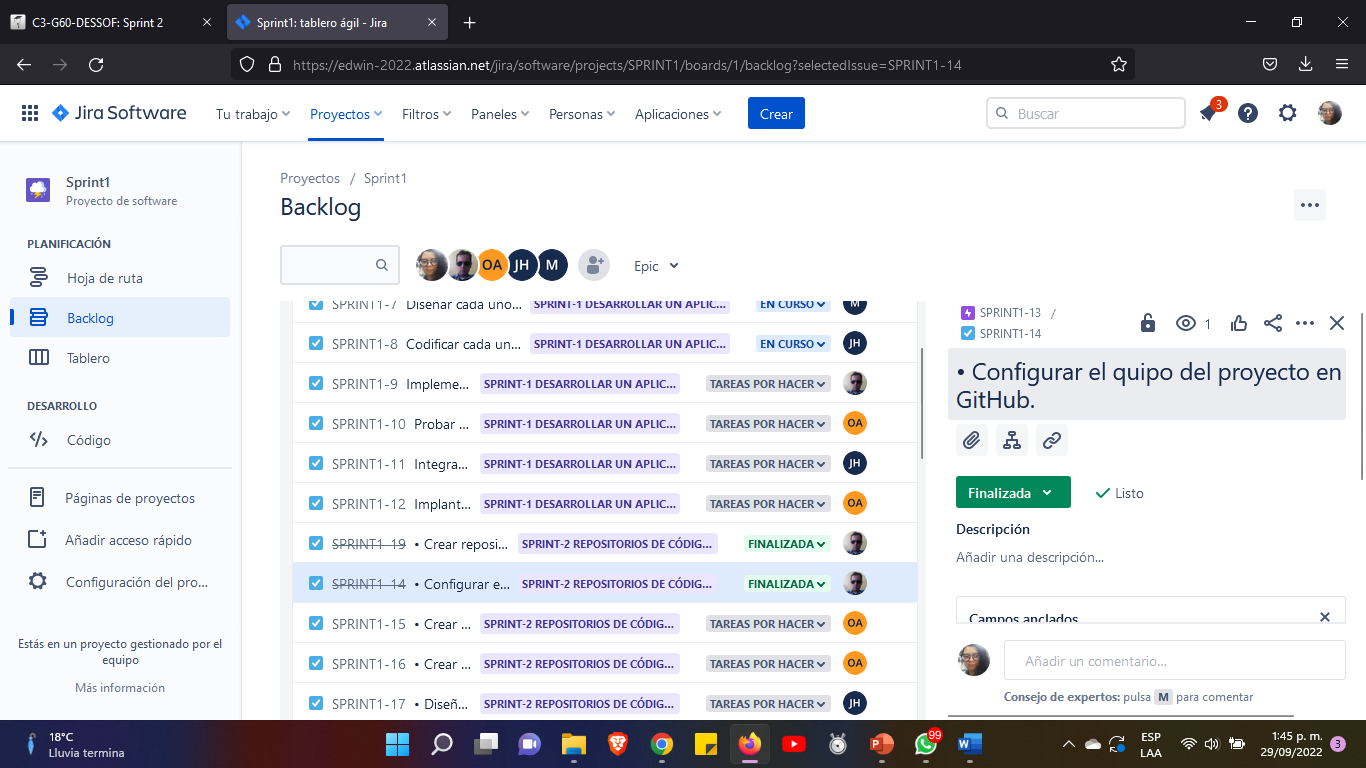
ON DELETE NO ACTION

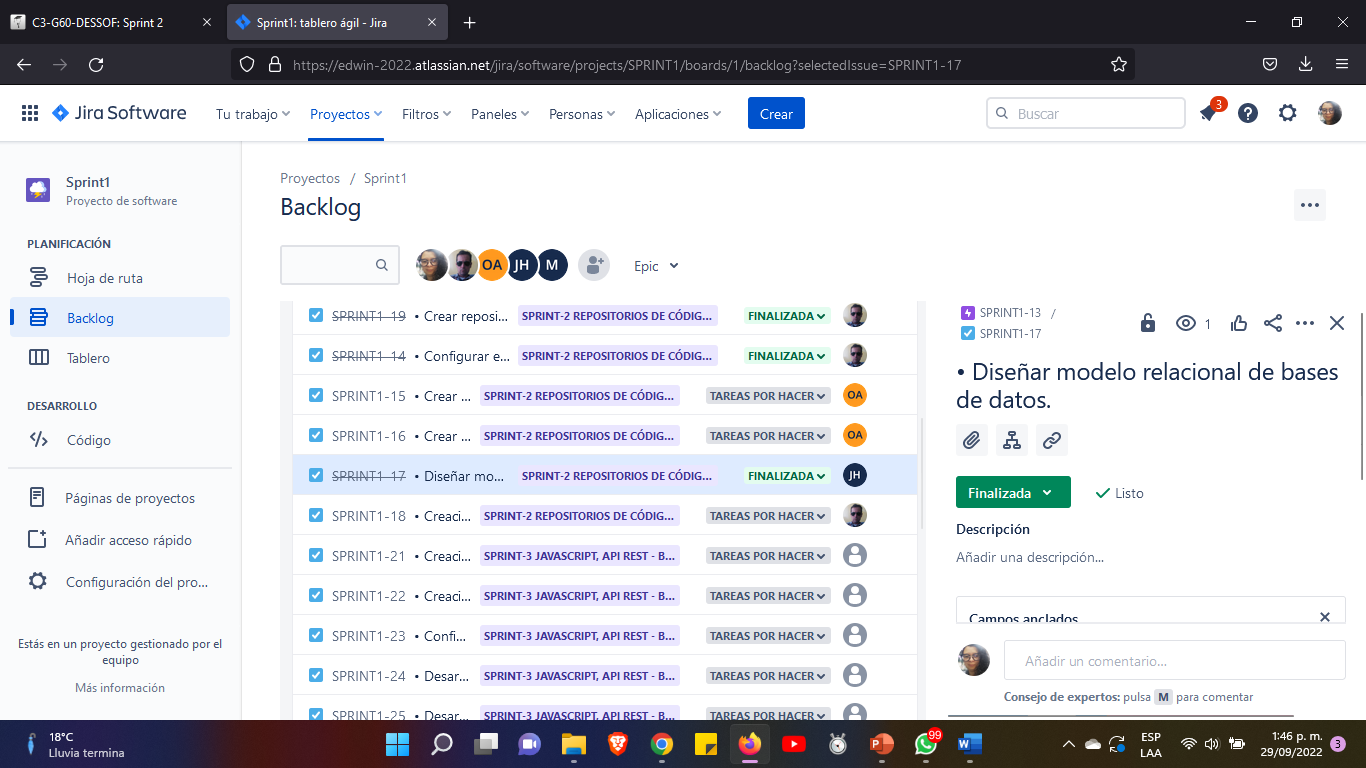
ON UPDATE NO ACTION);



| **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.

