SPRINT 2: Repositorio de Código - Diseño Base de Datos

Identificación Proyecto	
Nombre Proyecto:	ADMISTRACION.COM
Número Equipo:	3
Integrantes del equipo	
Rol (Líder-Desarrollador – Cliente)	Nombre
Lider	John Ferney Niño Alvarado
Desarrollador	Diego Mauricio Palacios
Desarrollador	Melvin Peñarada
Desarrollador	Deison Ruiz
Desarrollador	Elsyn Vargas

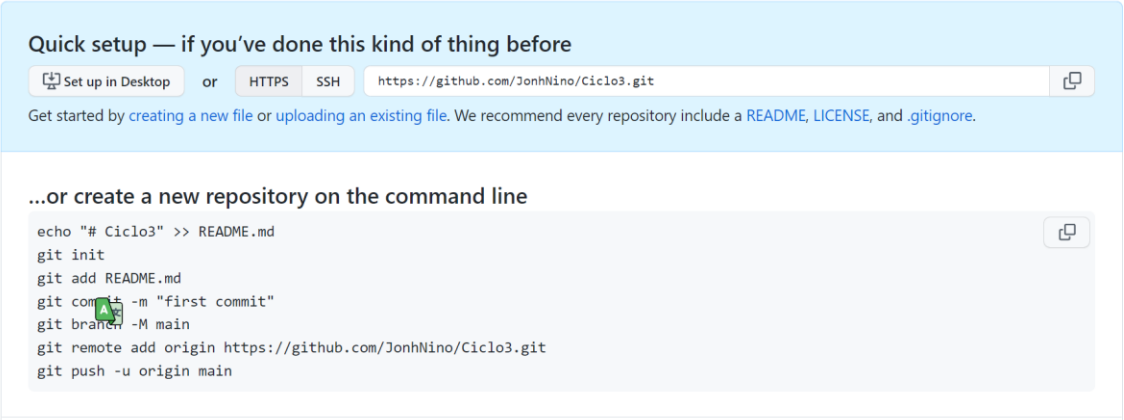
Repositorio de Código GitLab o GitHub

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)

Repositorio de Código GitHub

Repositorio: https://github.com/JonhNino/Ciclo3



Quick setup — if you've done this kind of thing before

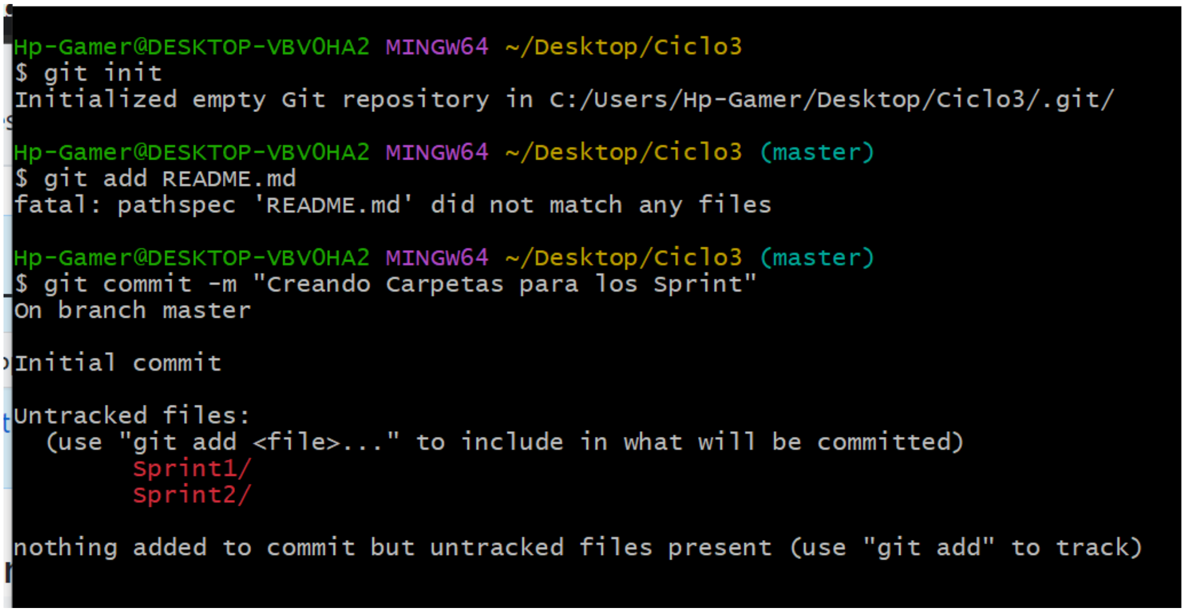
https://github.com/JonhNino/Ciclo3.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

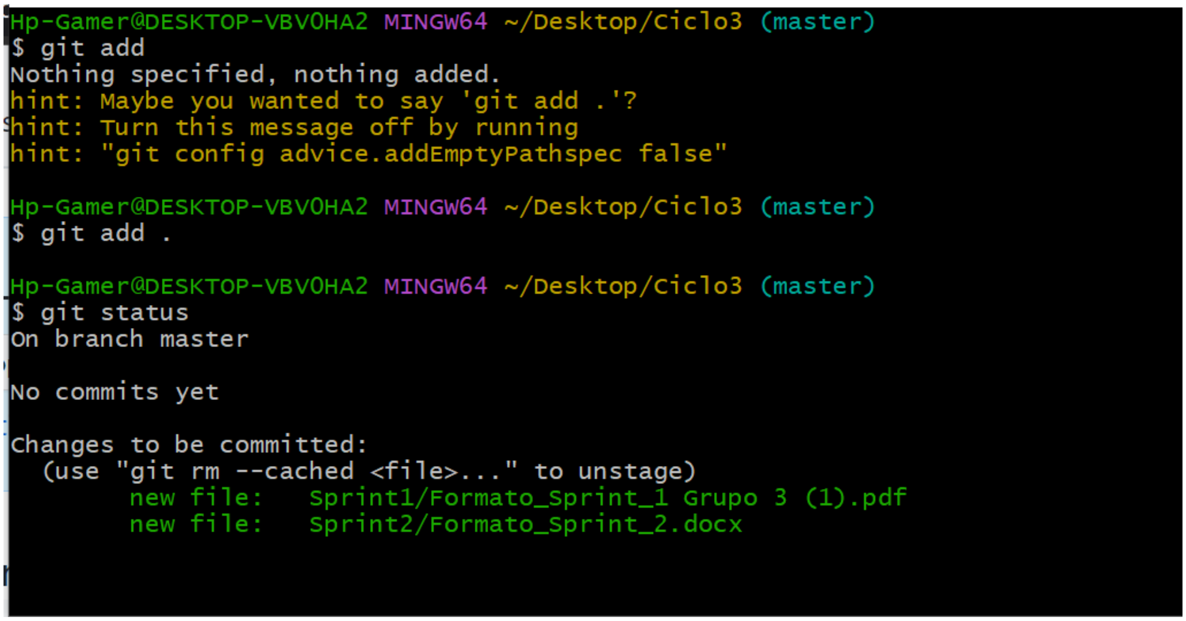
```
echo "# Ciclo3" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/JonhNino/Ciclo3.git
git push -u origin main
```

Fig. 1. Creación repositorio en la Plataforma Github.



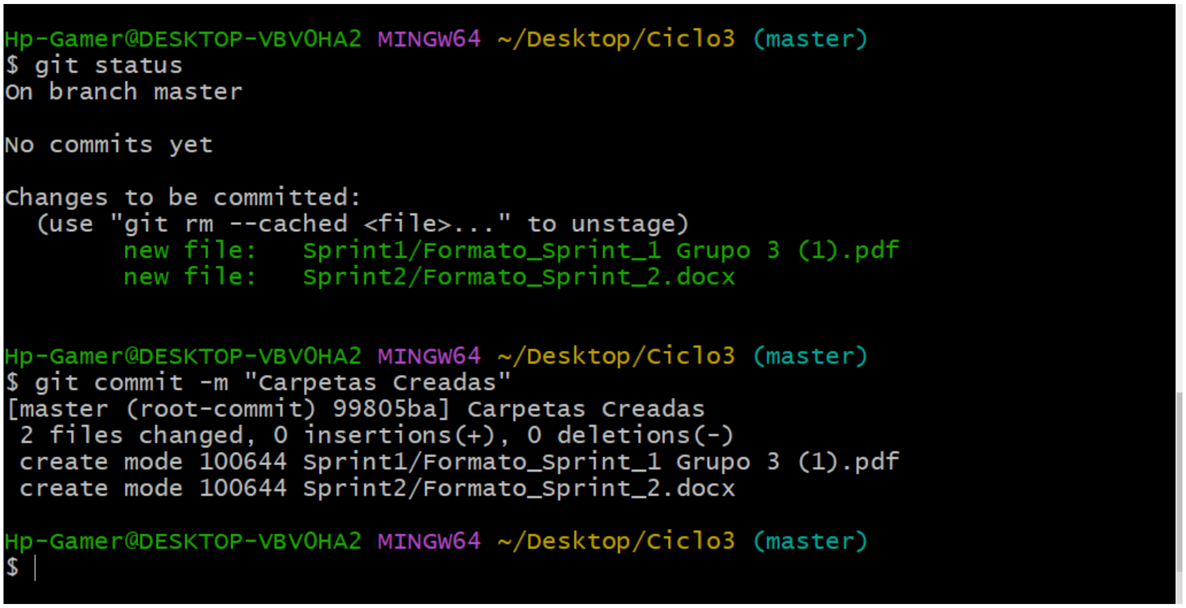
```
Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3
$ git init
Initialized empty Git repository in C:/Users/Hp-Gamer/Desktop/Ciclo3/.git/

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git add README.md
fatal: pathspec 'README.md' did not match any files

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git commit -m "Creando Carpetas para los Sprint"
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Sprint1/
        Sprint2/

nothing added to commit but untracked files present (use "git add" to track)
```

Fig. 2. Creación Repositorio Local.



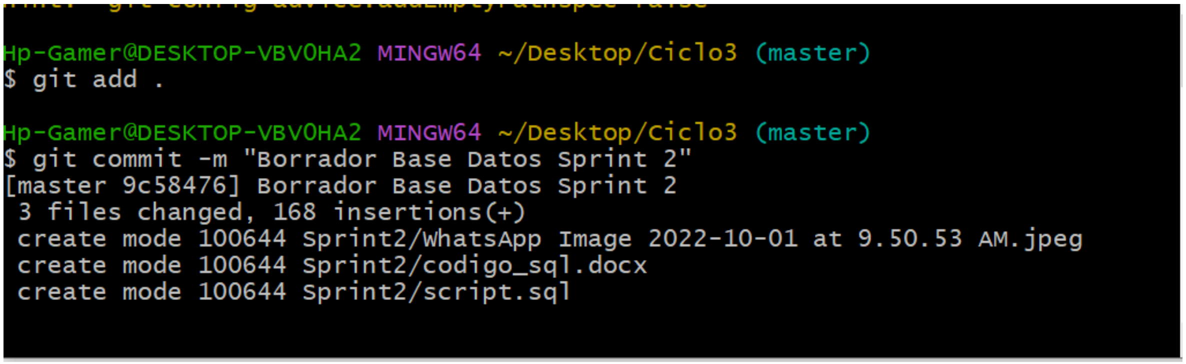
```
Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'?
hint: Turn this message off by running
hint: "git config advice.addEmptyPathspec false"

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git add .

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Sprint1/Formato_Sprint_1 Grupo 3 (1).pdf
        new file:   Sprint2/Formato_Sprint_2.docx
```

Fig. 3. Realización Git add.



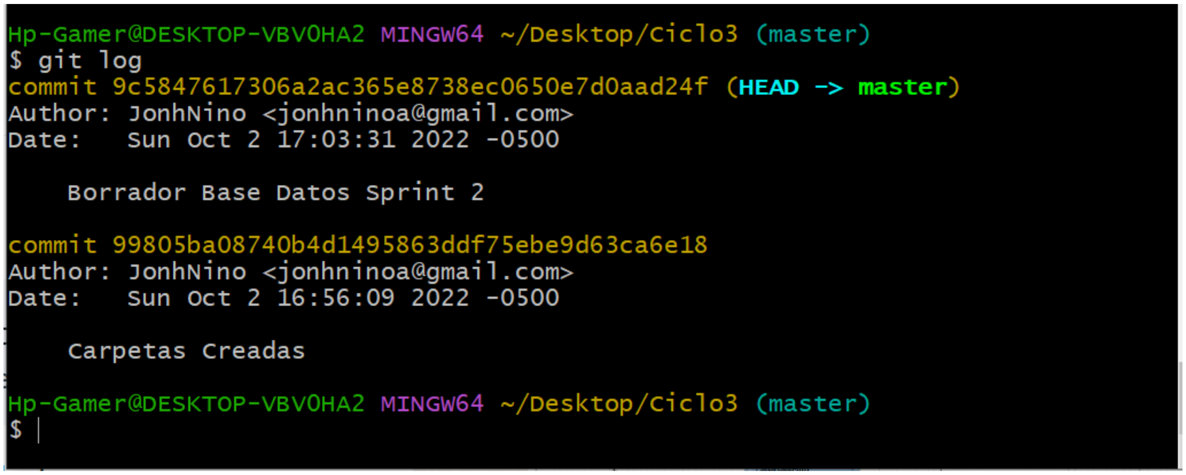
```
Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Sprint1/Formato_Sprint_1 Grupo 3 (1).pdf
        new file:   Sprint2/Formato_Sprint_2.docx


Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git commit -m "Carpetas Creadas"
[master (root-commit) 99805ba] Carpetas Creadas
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Sprint1/Formato_Sprint_1 Grupo 3 (1).pdf
 create mode 100644 Sprint2/Formato_Sprint_2.docx

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ |
```

Fig. 4. Realización Git Commit “Carpetas Creadas”.



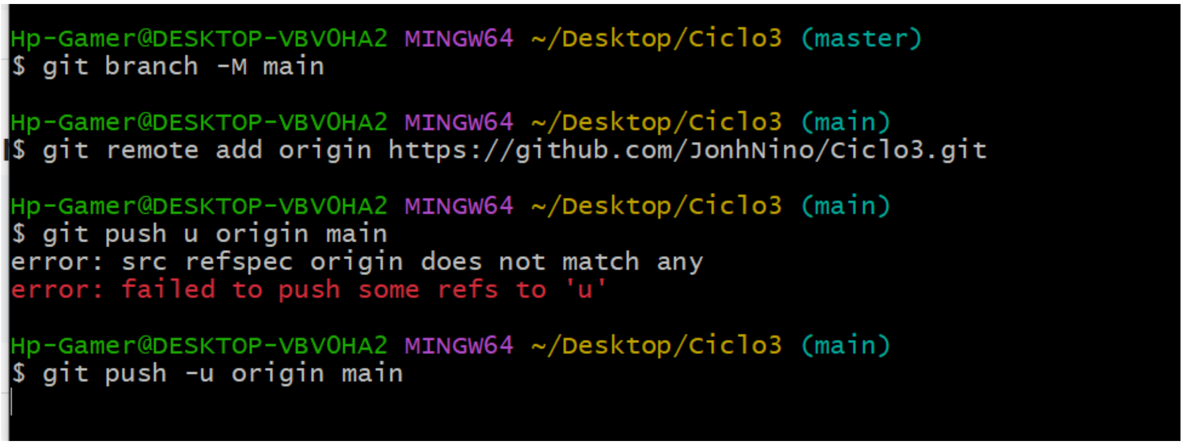
```
hint: git config advice.addEmptyPathspec false

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git add .

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git commit -m "Borrador Base Datos Sprint 2"
[master 9c58476] Borrador Base Datos Sprint 2
 3 files changed, 168 insertions(+)
 create mode 100644 Sprint2/WhatsApp Image 2022-10-01 at 9.50.53 AM.jpeg
 create mode 100644 Sprint2/codigo_sql.docx
 create mode 100644 Sprint2/script.sql
```

Fig. 5. Realización Git Commit “Borrador Base Datos sprint 2”.



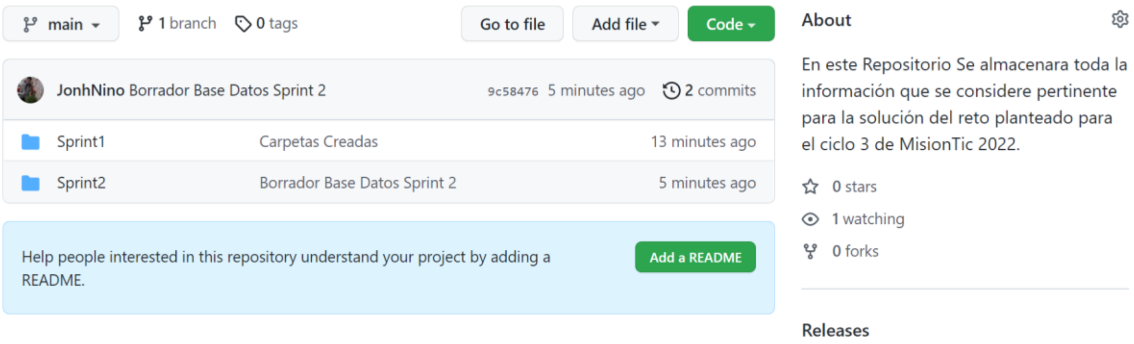
```
Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git log
commit 9c5847617306a2ac365e8738ec0650e7d0aad24f (HEAD -> master)
Author: JonhNino <jonhninoa@gmail.com>
Date:   Sun Oct 2 17:03:31 2022 -0500

    Borrador Base Datos Sprint 2

commit 99805ba08740b4d1495863ddf75ebe9d63ca6e18
Author: JonhNino <jonhninoa@gmail.com>
Date:   Sun Oct 2 16:56:09 2022 -0500

    Carpetas Creadas

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ |
```

Fig. 6. Se visualizan los Commit’s hechos hasta el momento.



```
Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (master)
$ git branch -M main

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (main)
$ git remote add origin https://github.com/JonhNino/Ciclo3.git

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (main)
$ git push u origin main
error: src refspec origin does not match any
error: failed to push some refs to 'u'

Hp-Gamer@DESKTOP-VBVOHA2 MINGW64 ~/Desktop/Ciclo3 (main)
$ git push -u origin main
```

Fig. 7. Realización Git Push https://github.com/JonhNino/Ciclo3 creado en la Plataforma Githb.



main 1 branch 0 tags Go to file Add file Code

JonhNino Borrador Base Datos Sprint 2 9c58476 5 minutes ago 2 commits

Sprint1	Carpetas Creadas	13 minutes ago
Sprint2	Borrador Base Datos Sprint 2	5 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

En este Repositorio Se almacenara toda la información que se considere pertinente para la solución del reto planteado para el ciclo 3 de MisionTic 2022.

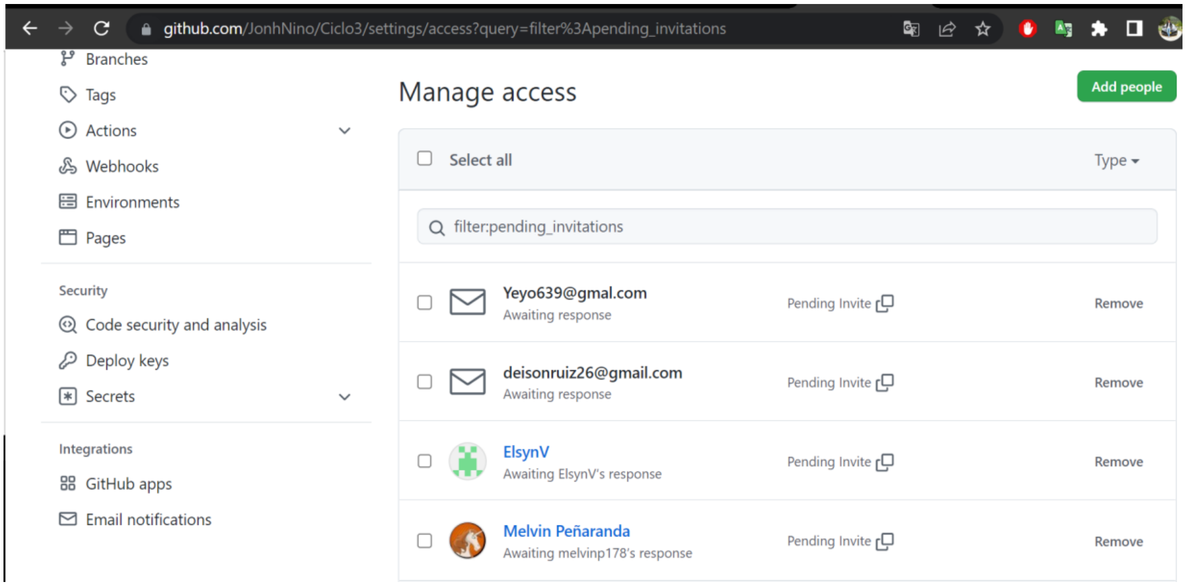
0 stars

1 watching

0 forks

Releases

Fig. 8. Captura del estado del repositorio creado en la plataforma github, luego de haber realizado el push en el repositorio Local.



github.com/JonhNino/Ciclo3/settings/access?query=filter%3Apending_invitations

Manage access

Add people

□ Select all Type

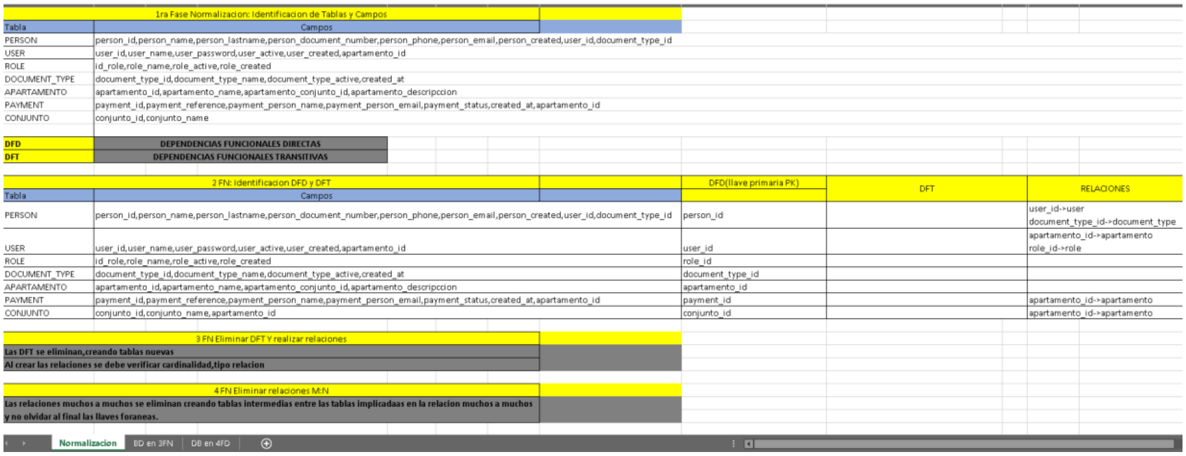
filter:pending_invitations

□	Yeyo639@gmal.com Awaiting response	Pending Invite	Remove
□	deisonruiz26@gmail.com Awaiting response	Pending Invite	Remove
□	ElsynV Awaiting ElsynV's response	Pending Invite	Remove
□	Melvin Peñaranda Awaiting melvinp178's response	Pending Invite	Remove

Fig. 9. Invitación a los Colaboradores del Proyecto.

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).



1ra Fase Normalizacion: Identificacion de Tablas y Campos	
Tabla	Campos
PERSON	person_id,person_name,person_lastname,person_document_number,person_phone,person_email,person_created,user_id,document_type_id
USER	user_id,user_name,user_password,user_active,user_created,apartamento_id
ROLE	id_role,role_name,role_active,role_created
DOCUMENT_TYPE	document_type_id,document_type_name,document_type_active,created_at
APARTAMENTO	apartamento_id,apartamento_name,apartamento_conjunto_id,apartamento_descripccion
PAYMENT	payment_id,payment_reference,payment_person_name,payment_person_email,payment_status,created_at,apartamento_id
CONJUNTO	conjunto_id,conjunto_name

DFD	DEPENDENCIAS FUNCIONALES DIRECTAS
DFT	DEPENDENCIAS FUNCIONALES TRANSITIVAS

2 FN: Identificacion DFD y DFT		DFD(llave primaria PK)	DFT	RELACIONES
Tabla	Campos			
PERSON	person_id,person_name,person_lastname,person_document_number,person_phone,person_email,person_created,user_id,document_type_id	person_id		user_id->user document_type_id->document_type
USER	user_id,user_name,user_password,user_active,user_created,apartamento_id	user_id		apartamento_id->apartamento role_id->role
ROLE	id_role,role_name,role_active,role_created	role_id		
DOCUMENT_TYPE	document_type_id,document_type_name,document_type_active,created_at	document_type_id		
APARTAMENTO	apartamento_id,apartamento_name,apartamento_conjunto_id,apartamento_descripccion	apartamento_id		
PAYMENT	payment_id,payment_reference,payment_person_name,payment_person_email,payment_status,created_at,apartamento_id	payment_id		apartamento_id->apartamento
CONJUNTO	conjunto_id,conjunto_name,apartamento_id	conjunto_id		apartamento_id->apartamento

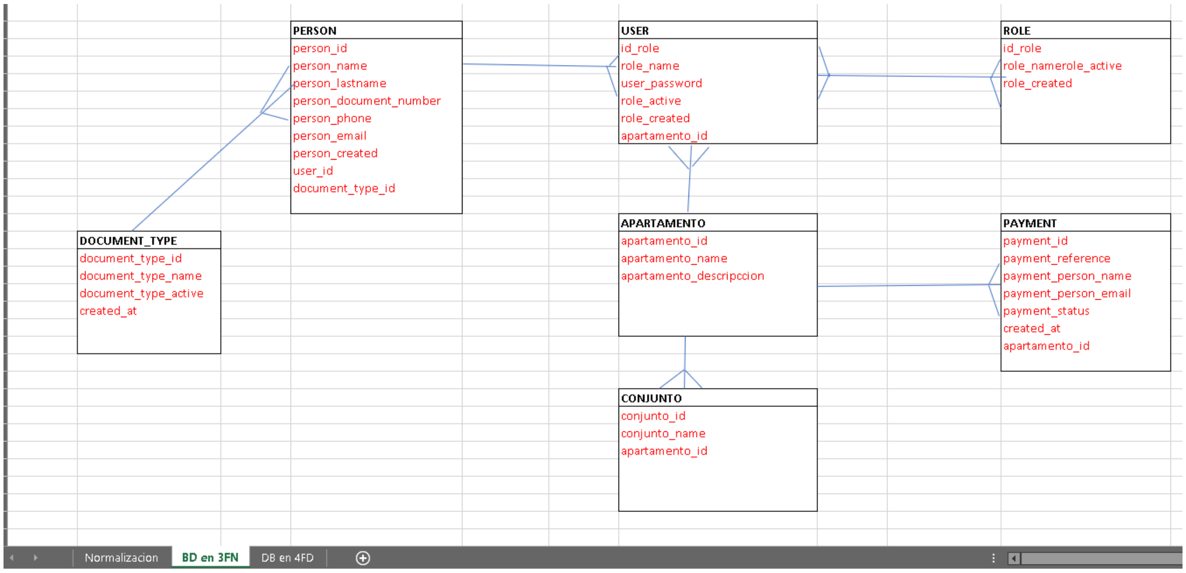
3 FN Eliminar DFT Y realizar relaciones

Las DFT se eliminan,creando tablas nuevas

Al crear las relaciones se debe verificar cardinalidad,tipo relacion

4 FN Eliminar relaciones M:N

Las relaciones muchos a muchos se eliminan creando tablas intermedias entre las tablas implicadaas en la relacion muchos a muchos y no olvidar al final las llaves foraneas.



PERSON
person_id
person_name
person_lastname
person_document_number
person_phone
person_email
person_created
user_id
document_type_id

USER
id_role
role_name
user_password
role_active
role_created
apartamento_id

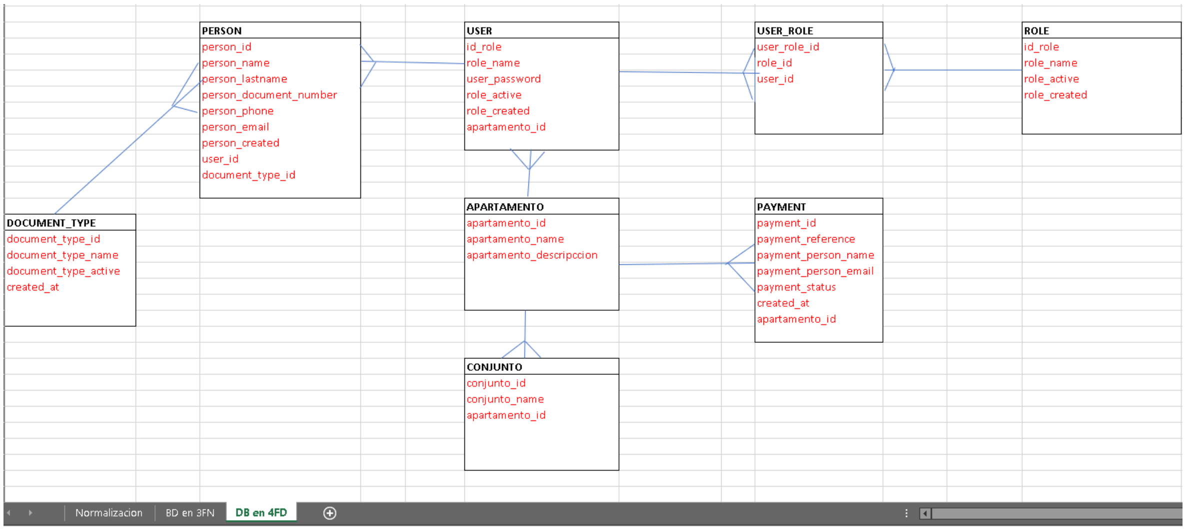
ROLE
id_role
role_namerole_active
role_created

DOCUMENT_TYPE
document_type_id
document_type_name
document_type_active
created_at

APARTAMENTO
apartamento_id
apartamento_name
apartamento_descripccion

PAYMENT
payment_id
payment_reference
payment_person_name
payment_person_email
payment_status
created_at
apartamento_id

CONJUNTO
conjunto_id
conjunto_name
apartamento_id


PERSON
person_id
person_name
person_lastname
person_document_number
person_phone
person_email
person_created
user_id
document_type_id

USER
id_role
role_name
user_password
role_active
role_created
apartamento_id

USER_ROLE
user_role_id
role_id
user_id

ROLE
id_role
role_name
role_active
role_created

DOCUMENT_TYPE
document_type_id
document_type_name
document_type_active
created_at

APARTAMENTO
apartamento_id
apartamento_name
apartamento_descripccion

PAYMENT
payment_id
payment_reference
payment_person_name
payment_person_email
payment_status
created_at
apartamento_id

CONJUNTO
conjunto_id
conjunto_name
apartamento_id

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.

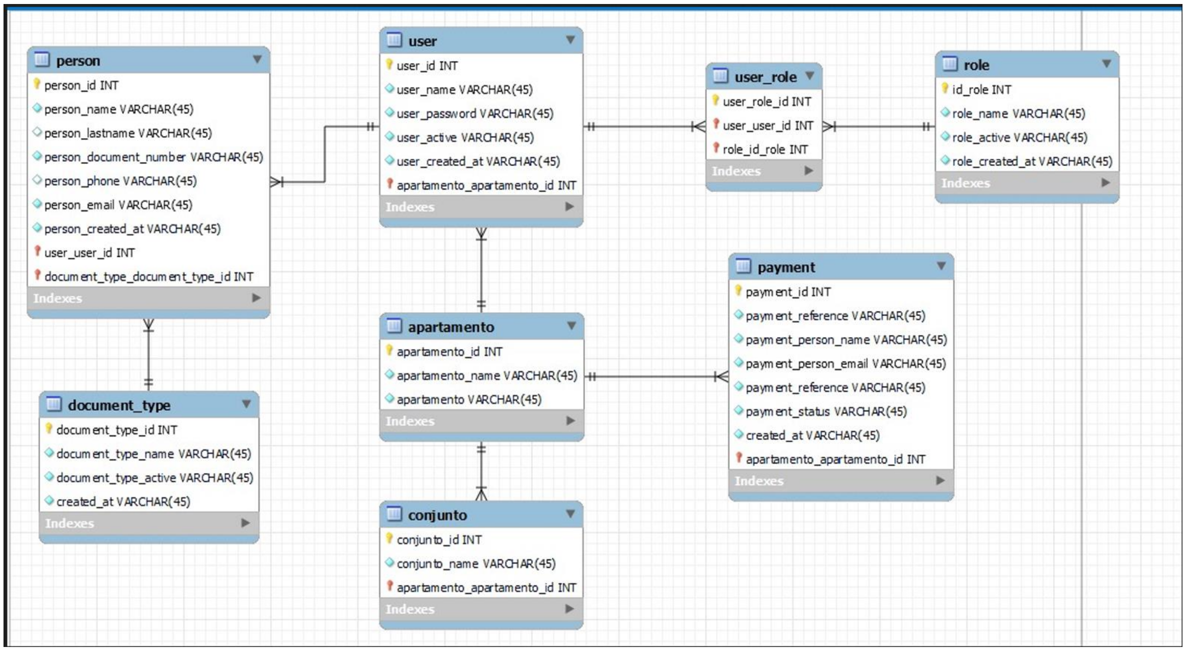
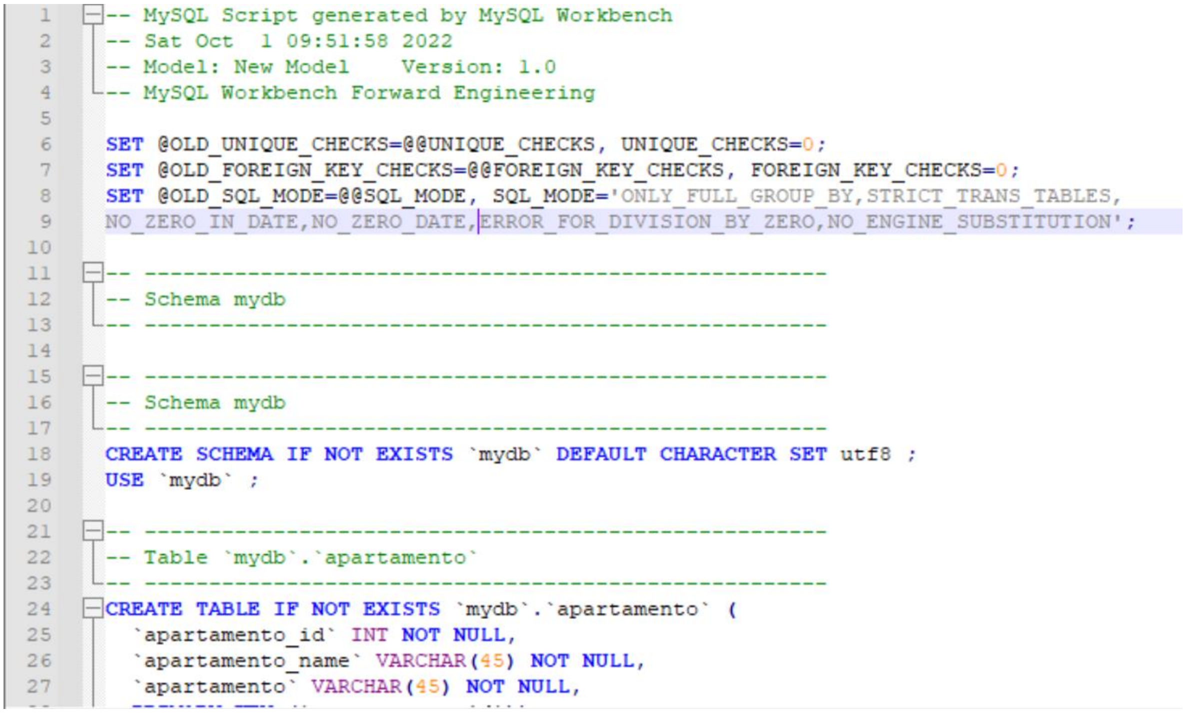
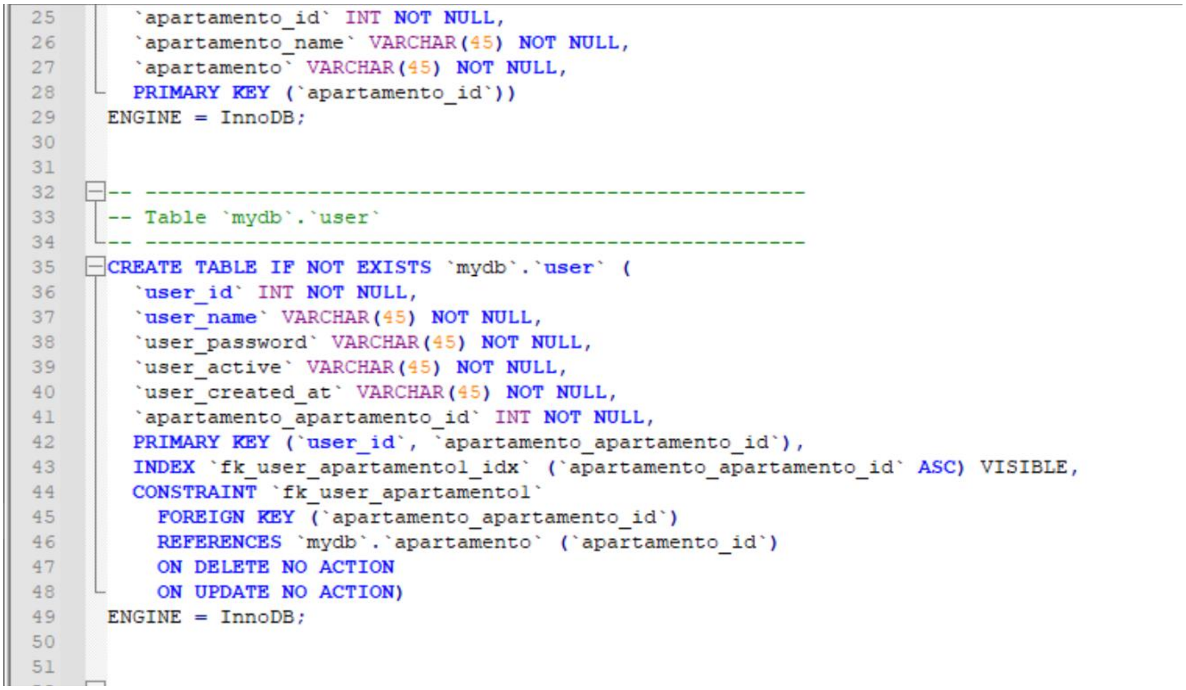

person
person_id INT
person_name VARCHAR(45)
person_lastname VARCHAR(45)
person_document_number VARCHAR(45)
person_phone VARCHAR(45)
person_email VARCHAR(45)
person_created_at VARCHAR(45)
user_user_id INT
document_type_document_type_id INT
Indexes
user
user_id INT
user_name VARCHAR(45)
user_password VARCHAR(45)
user_active VARCHAR(45)
user_created_at VARCHAR(45)
apartamento_apartamento_id INT
Indexes
user_role
user_role_id INT
user_user_id INT
role_id_role INT
Indexes
role
id_role INT
role_name VARCHAR(45)
role_active VARCHAR(45)
role_created_at VARCHAR(45)
Indexes
payment
payment_id INT
payment_reference VARCHAR(45)
payment_person_name VARCHAR(45)
payment_person_email VARCHAR(45)
payment_reference VARCHAR(45)
payment_status VARCHAR(45)
created_at VARCHAR(45)
apartamento_apartamento_id INT
Indexes
apartamento
apartamento_id INT
apartamento_name VARCHAR(45)
apartamento VARCHAR(45)
Indexes
document_type
document_type_id INT
document_type_name VARCHAR(45)
document_type_active VARCHAR(45)
created_at VARCHAR(45)
Indexes
conjunto
conjunto_id INT
conjunto_name VARCHAR(45)
apartamento_apartamento_id INT
Indexes

Fig. 10. Diagrama Entidad relación que se Utilizara para el Desarrollo.



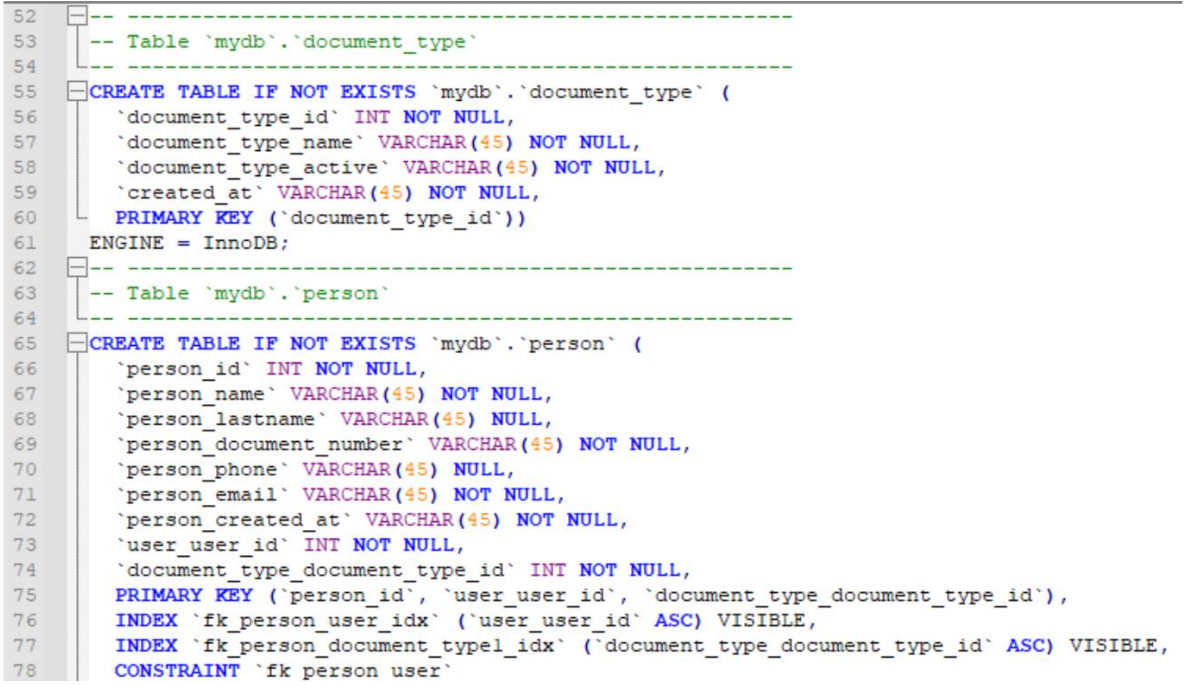
```
-- MySQL Script generated by MySQL Workbench
-- Sat Oct  1 09:51:58 2022
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

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`.`apartamento`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`apartamento` (
  `apartamento_id` INT NOT NULL,
  `apartamento_name` VARCHAR(45) NOT NULL,
  `apartamento` VARCHAR(45) NOT NULL,
```

Fig. 11. Configuración y Creación de la base de Datos a Usar.



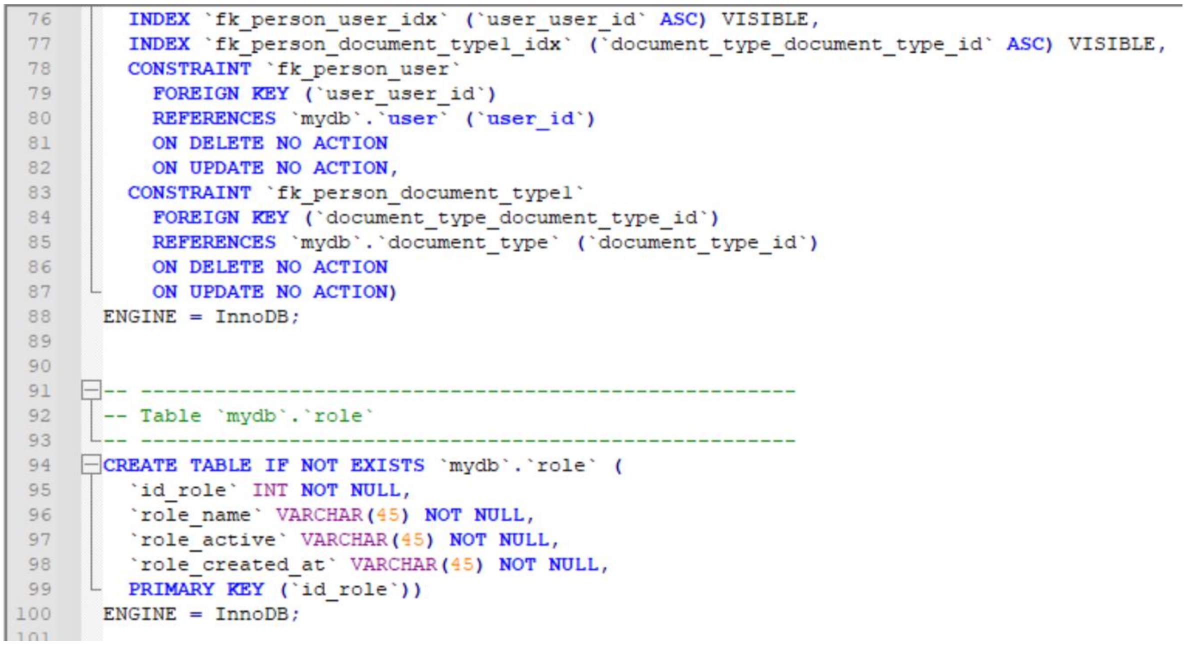
```
  `apartamento_id` INT NOT NULL,
  `apartamento_name` VARCHAR(45) NOT NULL,
  `apartamento` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`apartamento_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`user`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`user` (
  `user_id` INT NOT NULL,
  `user_name` VARCHAR(45) NOT NULL,
  `user_password` VARCHAR(45) NOT NULL,
  `user_active` VARCHAR(45) NOT NULL,
  `user_created_at` VARCHAR(45) NOT NULL,
  `apartamento_apartamento_id` INT NOT NULL,
  PRIMARY KEY (`user_id`, `apartamento_apartamento_id`),
  INDEX `fk_user_apartamento1_idx` (`apartamento_apartamento_id` ASC) VISIBLE,
  CONSTRAINT `fk_user_apartamento1`
    FOREIGN KEY (`apartamento_apartamento_id`)
    REFERENCES `mydb`.`apartamento` (`apartamento_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Fig. 12. Creación tablas Usuario y Apartamento, con sus respectivas Características (Variables y llaves Primarias y Foráneas).



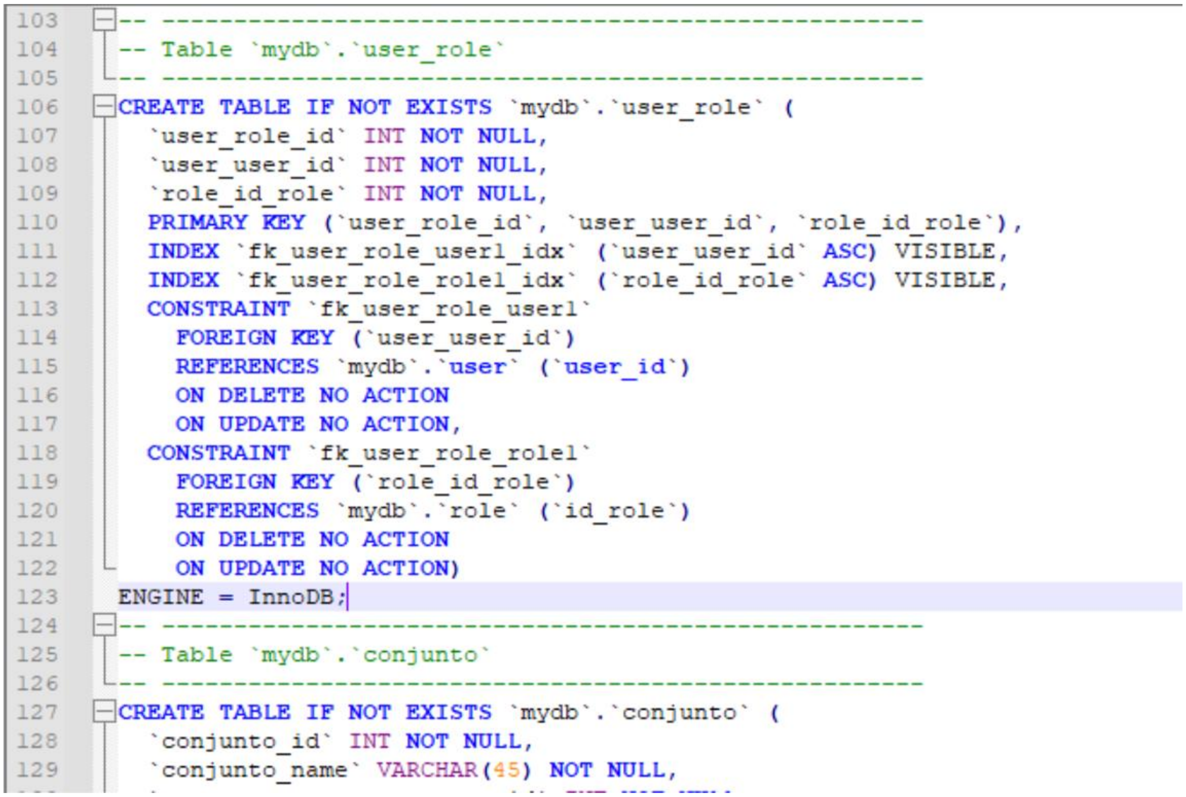
```
-- -----------------------------------------------------
-- Table `mydb`.`document_type`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`document_type` (
  `document_type_id` INT NOT NULL,
  `document_type_name` VARCHAR(45) NOT NULL,
  `document_type_active` VARCHAR(45) NOT NULL,
  `created_at` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`document_type_id`))
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `mydb`.`person`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`person` (
  `person_id` INT NOT NULL,
  `person_name` VARCHAR(45) NOT NULL,
  `person_lastname` VARCHAR(45) NULL,
  `person_document_number` VARCHAR(45) NOT NULL,
  `person_phone` VARCHAR(45) NULL,
  `person_email` VARCHAR(45) NOT NULL,
  `person_created_at` VARCHAR(45) NOT NULL,
  `user_user_id` INT NOT NULL,
  `document_type_document_type_id` INT NOT NULL,
  PRIMARY KEY (`person_id`, `user_user_id`, `document_type_document_type_id`),
  INDEX `fk_person_user_idx` (`user_user_id` ASC) VISIBLE,
  INDEX `fk_person_document_type1_idx` (`document_type_document_type_id` ASC) VISIBLE,
  CONSTRAINT `fk_person_user`
```

Fig. 13. Creación tablas Tipo Documento y Persona, con sus respectivas Características (Variables y llaves Primarias y Foráneas).



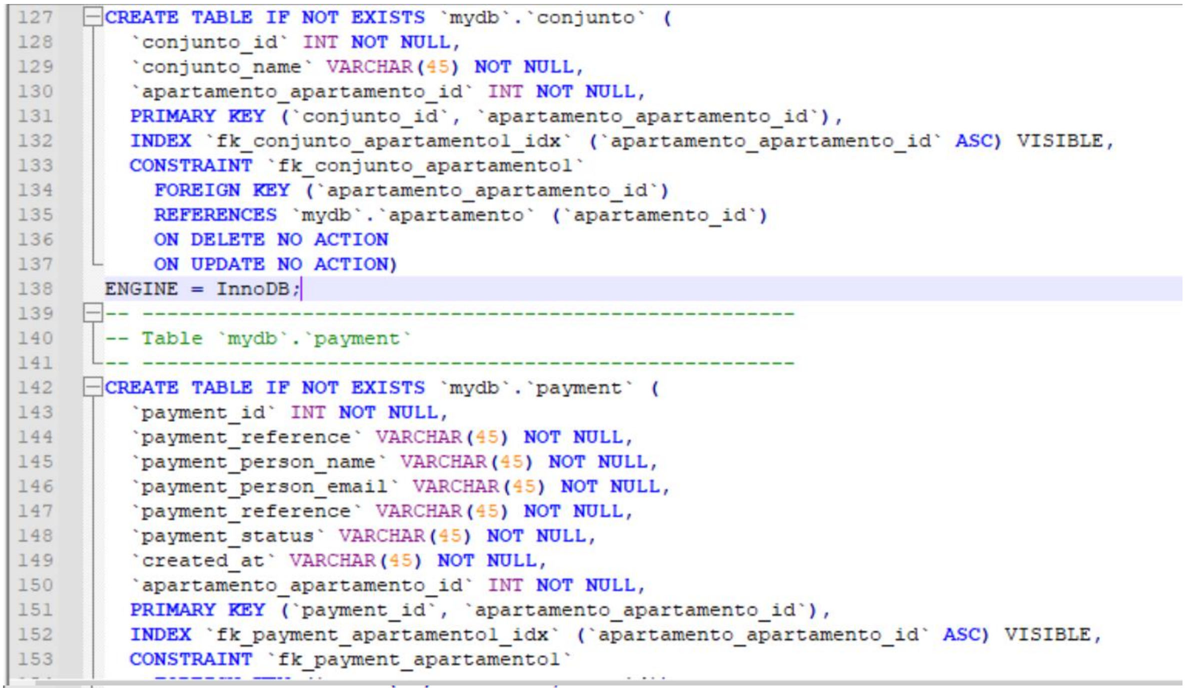
```
  INDEX `fk_person_user_idx` (`user_user_id` ASC) VISIBLE,
  INDEX `fk_person_document_type1_idx` (`document_type_document_type_id` ASC) VISIBLE,
  CONSTRAINT `fk_person_user`
    FOREIGN KEY (`user_user_id`)
    REFERENCES `mydb`.`user` (`user_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_person_document_type1`
    FOREIGN KEY (`document_type_document_type_id`)
    REFERENCES `mydb`.`document_type` (`document_type_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `mydb`.`role`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`role` (
  `id_role` INT NOT NULL,
  `role_name` VARCHAR(45) NOT NULL,
  `role_active` VARCHAR(45) NOT NULL,
  `role_created_at` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_role`))
ENGINE = InnoDB;
```

Fig. 14. Creación tabla Rol de usuario, con sus respectivas Características (Variables y llaves Primarias y Foráneas).

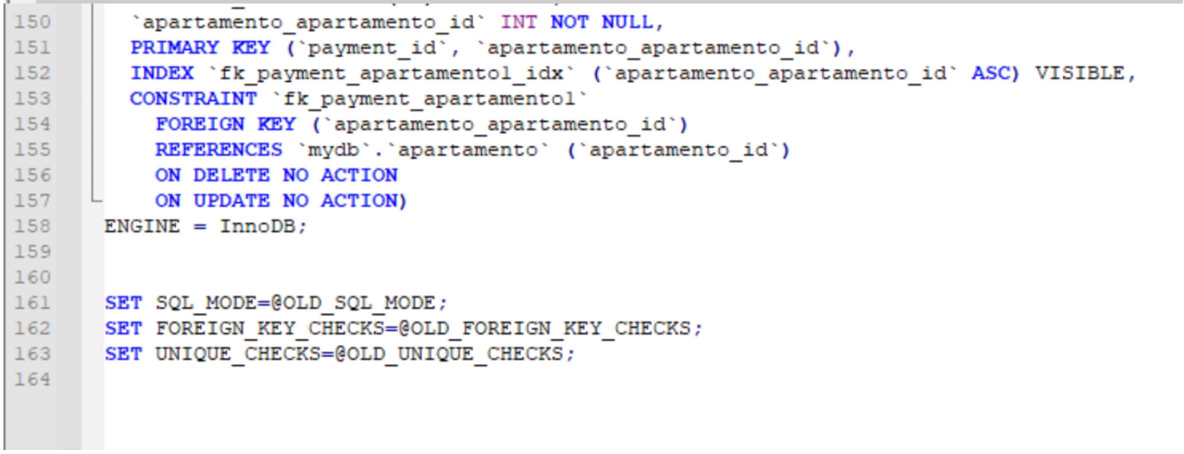


```
-- -----------------------------------------------------
-- Table `mydb`.`user_role`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`user_role` (
  `user_role_id` INT NOT NULL,
  `user_user_id` INT NOT NULL,
  `role_id_role` INT NOT NULL,
  PRIMARY KEY (`user_role_id`, `user_user_id`, `role_id_role`),
  INDEX `fk_user_role_user1_idx` (`user_user_id` ASC) VISIBLE,
  INDEX `fk_user_role_role1_idx` (`role_id_role` ASC) VISIBLE,
  CONSTRAINT `fk_user_role_user1`
    FOREIGN KEY (`user_user_id`)
    REFERENCES `mydb`.`user` (`user_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_role_role1`
    FOREIGN KEY (`role_id_role`)
    REFERENCES `mydb`.`role` (`id_role`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `mydb`.`conjunto`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`conjunto` (
  `conjunto_id` INT NOT NULL,
  `conjunto_name` VARCHAR(45) NOT NULL,
```

Fig. 15. Creación tablas Rol de usuario y Conjunto, con sus respectivas Características (Variables y llaves Primarias y Foráneas).



```sql
CREATE TABLE IF NOT EXISTS `mydb`.`conjunto` (
  `conjunto_id` INT NOT NULL,
  `conjunto_name` VARCHAR(45) NOT NULL,
  `apartamento_apartamento_id` INT NOT NULL,
  PRIMARY KEY (`conjunto_id`, `apartamento_apartamento_id`),
  INDEX `fk_conjunto_apartamentol_idx` (`apartamento_apartamento_id` ASC) VISIBLE,
  CONSTRAINT `fk_conjunto_apartamentol`
    FOREIGN KEY (`apartamento_apartamento_id`)
    REFERENCES `mydb`.`apartamento` (`apartamento_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `mydb`.`payment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `mydb`.`payment` (
  `payment_id` INT NOT NULL,
  `payment_reference` VARCHAR(45) NOT NULL,
  `payment_person_name` VARCHAR(45) NOT NULL,
  `payment_person_email` VARCHAR(45) NOT NULL,
  `payment_reference` VARCHAR(45) NOT NULL,
  `payment_status` VARCHAR(45) NOT NULL,
  `created_at` VARCHAR(45) NOT NULL,
  `apartamento_apartamento_id` INT NOT NULL,
  PRIMARY KEY (`payment_id`, `apartamento_apartamento_id`),
  INDEX `fk_payment_apartamentol_idx` (`apartamento_apartamento_id` ASC) VISIBLE,
  CONSTRAINT `fk_payment_apartamentol`
```

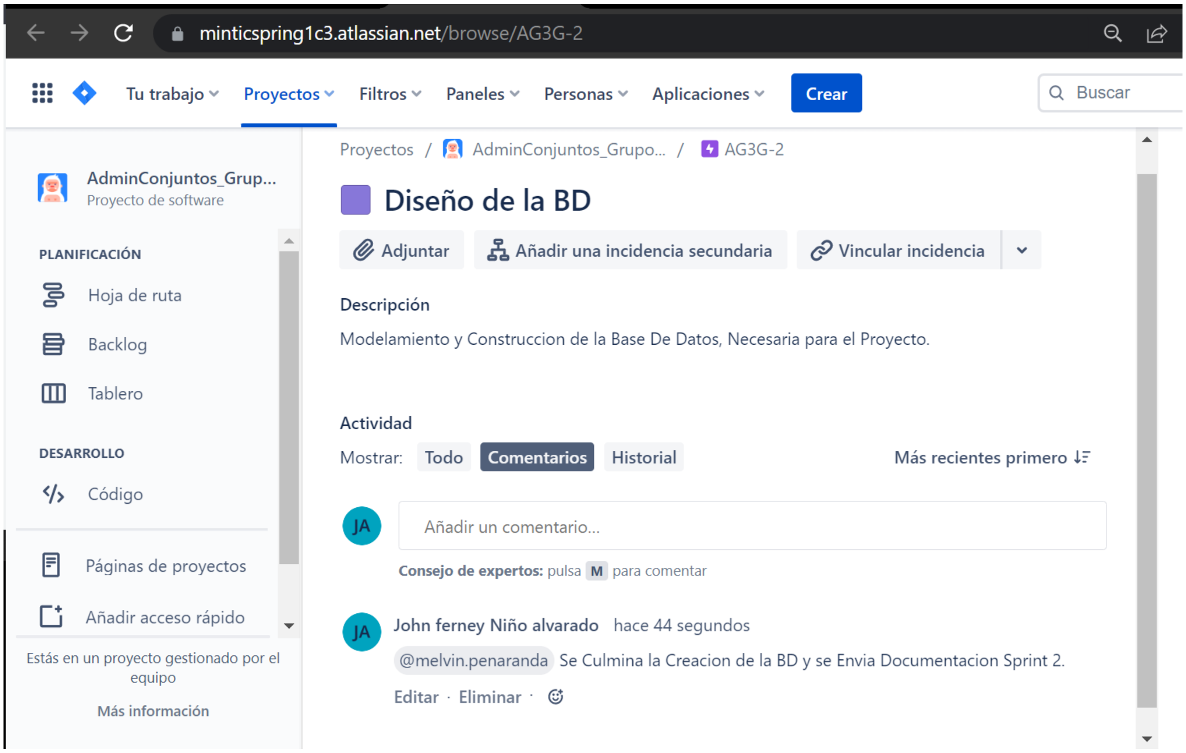


```sql
  `apartamento_apartamento_id` INT NOT NULL,
  PRIMARY KEY (`payment_id`, `apartamento_apartamento_id`),
  INDEX `fk_payment_apartamentol_idx` (`apartamento_apartamento_id` ASC) VISIBLE,
  CONSTRAINT `fk_payment_apartamentol`
    FOREIGN KEY (`apartamento_apartamento_id`)
    REFERENCES `mydb`.`apartamento` (`apartamento_id`)
    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;
```

Fig. 16 y 17. Creación tabla Rol de usuario, con sus respectivas Características (Variables y llaves Primarias y Foráneas).

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.



minticspring1c3.atlassian.net/browse/AG3G-2

Proyectos / AdminConjuntos_Grupo... / AG3G-2

AdminConjuntos_Grup...
Proyecto de software

Estás en un proyecto gestionado por el equipo

Más información

Diseño de la BD

Descripción

Modelamiento y Construccion de la Base De Datos, Necesaria para el Proyecto.

Actividad

Mostrar: Todo Comentarios Historial Más recientes primero

JA Añadir un comentario...

Consejo de expertos: pulsa M para comentar

JA John ferney Niño alvarado hace 44 segundos

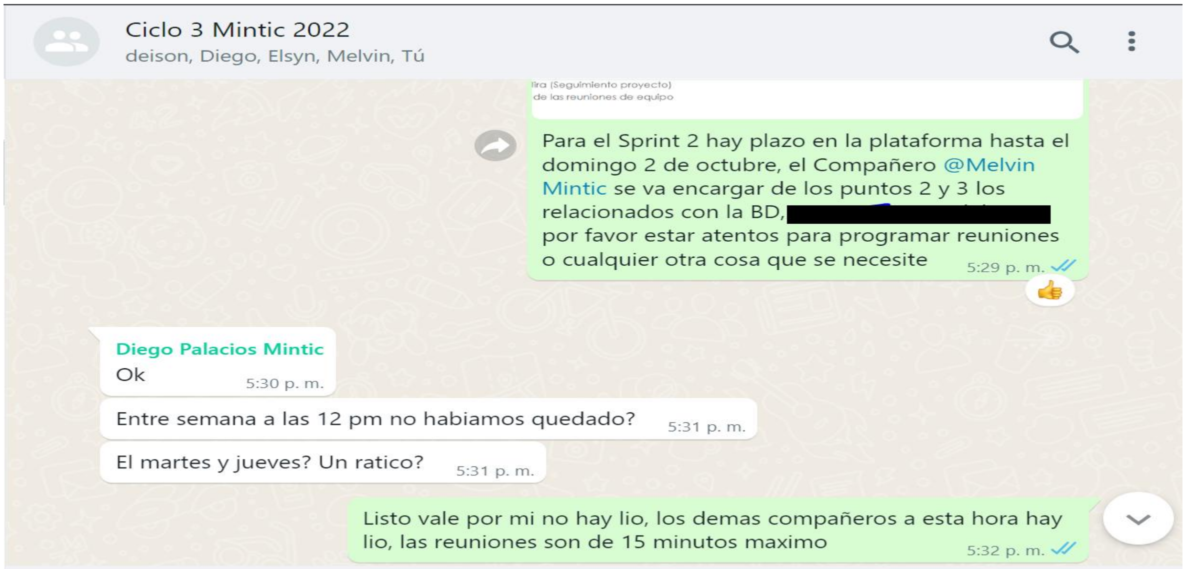
@melvin.penaranda Se Culmina la Creacion de la BD y se Envia Documentacion Sprint 2.

Editar · Eliminar ·

Fig. 18. Seguimiento Jira del requerimiento Diseño de la BD.

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.


Ciclo 3 Mintic 2022
deison, Diego, Elsyn, Melvin, Tú

lra (Seguimiento proyecto)
de las reuniones de equipo

Para el Sprint 2 hay plazo en la plataforma hasta el domingo 2 de octubre, el Compañero @Melvin Mintic se va encargar de los puntos 2 y 3 los relacionados con la BD, [redacted] por favor estar atentos para programar reuniones o cualquier otra cosa que se necesite 5:29 p. m.

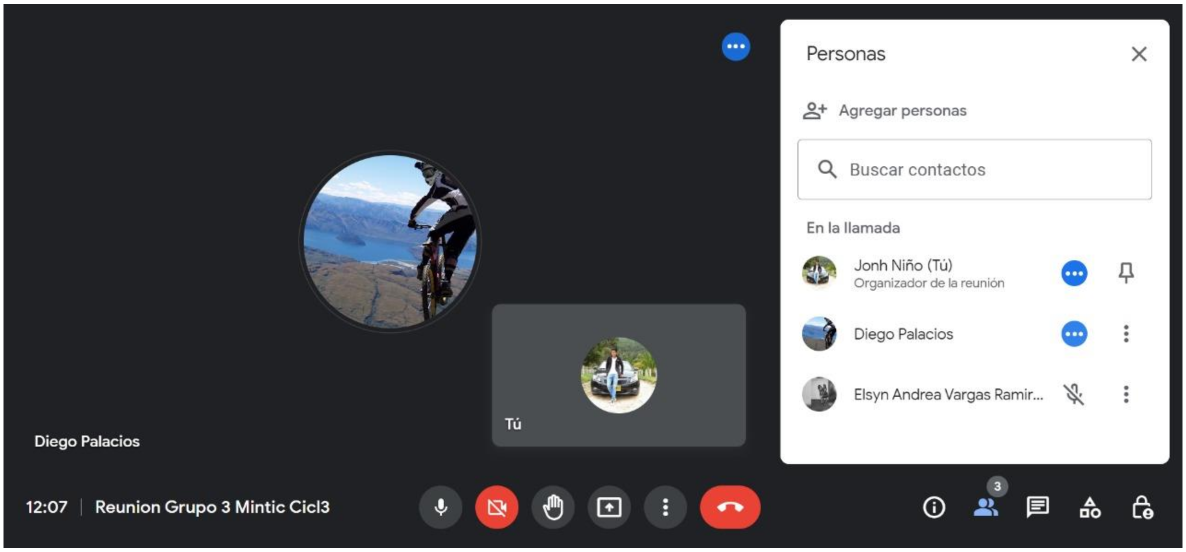
Diego Palacios Mintic
Ok 5:30 p. m.

Entre semana a las 12 pm no habiamos quedado? 5:31 p. m.

El martes y jueves? Un ratico? 5:31 p. m.

Listo vale por mi no hay lio, los demas compañeros a esta hora hay lio, las reuniones son de 15 minutos maximo 5:32 p. m.

Fig. 19. Captura de WhastApp, donde se muestra la asignación de tareas y programación de reunión virtual.


Personas

Agregar personas

Buscar contactos

En la llamada

Jonh Niño (Tú)
Organizador de la reunión

Diego Palacios

Elsyn Andrea Vargas Ramir...

Tú

Diego Palacios

12:07 | Reunion Grupo 3 Mintic Cicl3

Fig. 20. Captura Reunión Virtual vía Meet.